Karlsruhe Institute of Technology

Forest Stack Layouts

Bachelor's Thesis of

Lena Scherzer

At the Department of Informatics
Institute of Theoretical Informatics

Reviewers:	Dr. Torsten Ueckerdt
	T.T.-Prof. Thomas Bläsius
Advisors:	Laura Merker

Time Period: 30.05.2022 – 30.09.2022

Statement of Authorship

Ich versichere wahrheitsgemäß, die Arbeit selbstständig verfasst, alle benutzten Hilfsmittel vollständig und genau angegeben und alles kenntlich gemacht zu haben, was aus Arbeiten anderer unverändert oder mit Abänderungen entnommen wurde sowie die Satzung des KIT zur Sicherung guter wissenschaftlicher Praxis in der jeweils gültigen Fassung beachtet zu haben.

Karlsruhe, September 9, 2022

Abstract

A *stack layout* of a graph consists of an ordering of the vertices σ and a partition of the edges into *stacks*. Two edges on the same stack are not allowed to cross with respect to σ, meaning that for no two edges xy and uv with $\sigma(x) < \sigma(y)$ and $\sigma(u) < \sigma(v)$ in the same stack we have $\sigma(x) < \sigma(u) < \sigma(y) < \sigma(v)$ or $\sigma(u) < \sigma(x) < \sigma(v) < \sigma(y)$. In a *forest stack layout* the subgraph on each stack is a forest. The minimum number of stacks needed for a stack layout of a graph is called the *stack number* and the minimum number of stacks needed for a forest stack layout is called the *forest stack number*. We determine the forest stack number of some graphs and attack the conjecture that the forest stack number of graphs might be at most one greater than the stack number for every graph.

For complete graphs, complete bipartite graphs $K_{m,n}$ with $m \ll n$, planar 3-trees and k-trees the forest stack number or the best known bound on the forest stack number is shown to be the same as the stack number. For outerplanar graphs, one more stack is needed for forest stack layouts. We also show that for every k there is a graph with stack number k that needs $k+1$ stacks for a forest stack layout. Assuming some restrictions apply, we find counterexamples to the conjecture.

Deutsche Zusammenfassung

Ein *Stack Layout* eines Graphen besteht aus einer Reihenfolge der Knoten σ und einer Partition, welche die Kanten in *Stacks* aufteilt. Keine zwei Kanten in einem Stack dürfen sich kreuzen mit Bezug auf σ. Das bedeutet, dass für keine zwei Kanten uv, xy in einem Stack mit $\sigma(u) < \sigma(v)$ und $\sigma(x) < \sigma(y)$ gelten darf, dass $\sigma(x) < \sigma(u) < \sigma(y) < \sigma(v)$ oder $\sigma(u) < \sigma(x) < \sigma(v) < \sigma(y)$. Bei *Forest Stack Layouts* muss der Graph auf jedem Stack zusätzlich ein Wald sein. Die minimale Anzahl an Stacks, welche für ein Stack Layout eines Graphen benötigt werden, nennt sich die *Stack Nummer* und die minimale Anzahl an Stacks, welche für ein Forest Stack Layout eines Graphen benötigt werden, nennt sich die *Forest Stack Nummer*. Wir bestimmen die Forest Stack Nummer für einige Graphen und untersuchen die Vermutung, dass Forest Stack Layouts immer höchstens einen Stack mehr benötigen als Stack Layouts eines Graphen.

Für vollständige Graphen, vollständige bipartite Graphen $K_{m,n}$ mit $m \ll n$, planare 3-Bäume und k-Bäume zeigen wir, dass die Forest Stack Nummern bzw. die besten bekannten Schranken der Forest Stack Nummer die gleichen sind wie bei der Stack Nummer. Für außenplanare Graphen wird ein Stack mehr für ein Forest Stack Layout benötigt als für ein Stack Layout. Wir zeigen auch, dass für jedes k ein Graph mit Stack Nummer k und Forest Stack Nummer $k+1$ existiert. Wenn wir zusätzlich verschiedene Einschränkungen für die Layouts annehmen, finden wir Gegenbeispiele für die obige Vermutung.

Contents

1. Introduction

A *stack layout* of a graph is a partition $\mathcal{S}$ of the edges into sets, called stacks, together with an order σ of vertices. The idea is that the edges in each stack S_i in a stack layout can be inserted into and removed again from a stack using the given order of vertices σ. An edge is pushed on the stack when the first vertex adjacent to it is reached in σ. The edge is then removed when the second vertex of the edge is reached. As an example in Figure 1.1 the four edges of the shown graph can be put in the same stack since they can be added and removed from a stack as shown in Figure 1.1. When more than one edge is adjacent to a vertex, the correct order to push and pop the edges from the stack has to be chosen. A formal definition of stack layouts is given in Section 2.2.

Pushing and popping edges from a stack corresponds to the idea that on each stack no two edges cross with regards to the order of vertices σ. A different way of visualizing stack layouts is to search for a layout of a graph in a *book*. A book consists of half-planes, called pages, that intersect at the *spine* of the book. The vertices are placed on the spine of the book, creating the order of vertices σ. The edges are placed on the pages where no two edges are allowed to cross. An example of two crossing edges is given in Figure 1.3. Thus, stack layouts are also known as book embeddings. An example of a book embedding of a K_4 is shown in Figure 1.2.

In a stack layout each stack contains an outerplanar graph. Thus, the number of edges on each stack is limited.

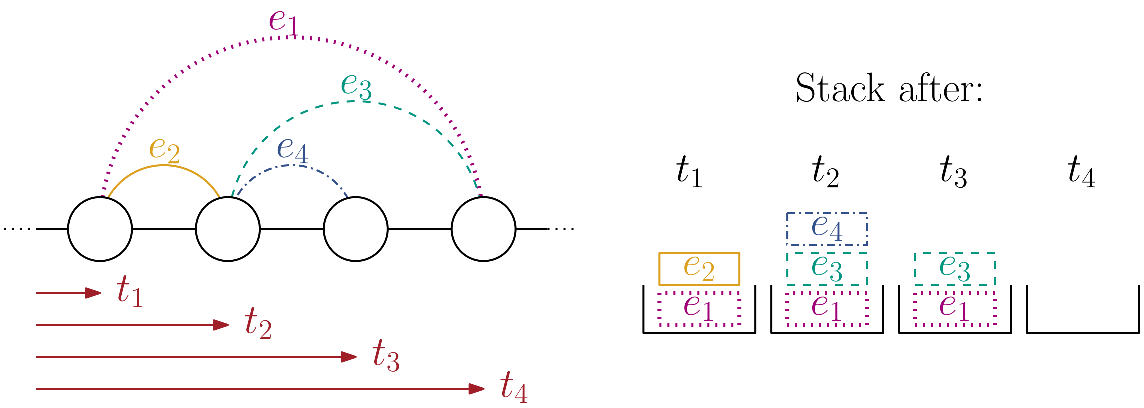

Figure 1.1: An example stack layout where all edges can be put in the same stack.

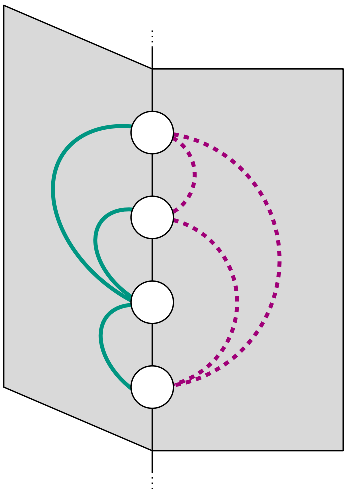

Figure 1.2: A book embedding of a K_4 on two pages.

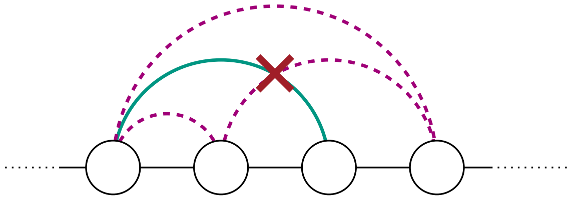

Figure 1.3: The green edge cannot be added to the purple page since it crosses another edge.

In this thesis, we consider a variant of stack layouts. *Forest stack layouts* have the additional property that the edges in each stack do not form cycles. Each stack in a forest stack layout thus induces a forest. This further limits the number of edges on each stack. A formal definition of forest stack layouts is given in Section 2.2.

A graph G has many different stack and forest stack layouts. The stack number $\mathrm{sn}(G)$ and forest stack number $\mathrm{sn_f}(G)$, however, is the smallest number of stacks needed to find a corresponding layout for the graph G. The stack number is also called the *page number* or *book thickness* of a graph.

Although stack layouts and the stack numbers of graphs have been more widely researched, not much is known about forest stack layouts and forest stack numbers. Therefore, we determine the forest stack number for some graphs for which the stack number is known. Additionally, we look at more general bounds and relations between the stack number and the forest stack number of graphs.

1.1 Motivation

The main motivation for this thesis is the following conjecture proposed by André Schulz and Jonathan Rollin.

Let $\mathrm{sn}(G) = k$ for a graph G with $|E(G)| = m$ and $|V(G)| = n$. That is, there exists a stack layout $\Gamma = (\sigma, \mathcal{S})$ of G with k stacks. The subgraph on each stack is an outerplanar graph and therefore contains at most $2n - 3$ edges. It follows that the total number of edges m is at most $k \cdot (2n - 3)$.

The edges between two vertices that are next to each other in the order of vertices σ are called *short edges*. Additionally, the edge between the first and last vertex in σ is also a short edge. Because the up to n short edges are counted on every stack of the layout, the bound for the number of edges can be reduced to $m \leq k(n-3) + n$.

Because $n - 2k \leq n - 1$ for $k \geq 1$ we can derive that

$$m \leq k(n-3) + n = k(n-1) + n - 2k \leq k(n-1) + (n-1) = (k+1)(n-1).$$

A stack in a forest stack layout can contain at most $n - 1$ edges. Therefore, the above inequality suggests that the following conjecture might hold.

Conjecture 1.1. *For a graph G with* $\mathrm{sn}(G) = k$, $\mathrm{sn_f}(G) \leq k+1$.

While forest stack layouts and the above conjecture might be interesting in and of themselves, they are also meaningful because they deepen our understanding of stack layouts. Additionally, when looking at stack layouts that use the minimum number of stacks it can be observed that often the stacks are already forests. If they are not already forests they are mostly quite sparse and can be changed into forests. This poses the question if this is only a coincidence or follows from Conjecture 1.1.

1.2 Related Work

The stack number of various graphs has been well researched. Here are some of the most important results on the stack number of graphs.

While the idea of embedding graphs in books existed before, the stack number of graphs was formalized by Bernhart and Kainen [3]. They also observed that graphs with stack number 1 are exactly outerplanar graphs and graphs with stack number at most 2 are exactly subgraphs of planar graphs with Hamiltonian cycles, so-called subhamiltonian graphs. In the same paper, they determined the stack number of complete graphs with more than three vertices to be $\mathrm{sn}(K_m) = \lceil m/2 \rceil$. For complete bipartite graphs $K_{m,n}$, they found the stack number to be m in the case that n is significantly larger than m.

For the stack number of complete bipartite graphs in general Muder et al.[16] proved the current best upper bound $\mathrm{sn}(K_{m,n}) \leq \lceil (m+2n)/4 \rceil$. For the special cases $K_{n,n}$ and $K_{\lfloor n^2/4 \rfloor, n}$ Enomoto et al.[5] proved that $\mathrm{sn}(K_{n,n}) \leq \lfloor 2n/3 \rfloor + 1$ and $\mathrm{sn}(K_{\lfloor n^2/4 \rfloor, n}) \leq n-1$.

Others have continued their work and characterized the stack number of different graph families. For example, Heath found an upper bound of 3 for the stack number of planar 3-trees[9].

Ganley and Heath proved that k-trees have at most stack number $k+1$[13]. Then Vandenbussche et al. showed that there are indeed k-trees that need $k+1$ stacks for a stack layout[22].

For planar graphs, Yannakakis proved that at most four stacks are needed for a stack layout[24]. Recently it was shown by Bekos et al. and Yannakakis independently that this boundary is tight and four stacks are necessary[2, 25].

Besides normal stack layouts, different variations of stack layouts have been defined and researched.

Overbay[19] considered different stack layouts where for example each stack is a cylinder. Pupyrev introduced the concept of simultaneous stack-queue layouts where for a shared order of vertices σ the edges are partitioned into a stack and a queue[20]. Whereas in stack layouts the edges are partitioned such that each set in the partition can be placed on a stack according to the order of vertices σ, in queue layouts the edges can be enqueued and dequeued with a first-in-first-out order σ.

Another example is *matching stack layouts*, which are stack layouts where the edges on each stack are a matching. These matching stack layouts are also called *dispersable book embeddings* and were also first introduced by Bernhart and Kainen [3]. These layouts are of additional interest to us because a matching stack layout is also a forest stack layout. Therefore, it can be said that forest stack layouts impose restrictions that are somewhere in between stack layouts and matching stack layouts.

Bernhart and Kainen conjectured that any k-regular bipartite graph has a matching stack layout using k stacks[3]. This conjecture was recently disproven by Alam et al.[1]. They were, however, able to prove that 3-regular bipartite planar graphs have matching stack layouts using three stacks.

Galil et al. call graphs that have matching k-stack layouts *k-pushdown graphs*. They were able to prove a minimum size for the smallest separator in large enough k-pushdown graphs[6].

Shao et al. [21] looked at the matching stack number of the Cartesian product of complete graphs and cycles. The Cartesian product C of two graphs G, H has as vertices the Cartesian product of the previous vertices $V(C) = V(G) \times V(H)$. Two vertices (u, v) and (u', v') are adjacent if and only if $u = u'$ and v was adjacent to v' in H or $v = v'$ and u was adjacent to u' in G. They proved that for those Cartesian products the matching stack number is the maximum degree plus one.

Forest stack layouts have also been previously defined and used by Merker and Ueckerdt[15]. They define the local stack number of a stack layout as the maximum number of stacks adjacent to any vertex. The local stack number of a graph is the minimum local stack number of any stack layout. They show that for k-trees the restriction to forest stacks does not change the local stack number.

For directed acyclic graphs, stack layouts have to fulfill an additional property. The order of vertices σ used in the layout has to be a topological ordering. Therefore, depending on the edges the possible orders σ are very limited. It was proven by Nowakowski and Parker and independently by Heath et al. that directed trees need only one stack for a stack layout[18, 11]. Additionally, directed acyclic graphs, where the corresponding undirected graph contains only one cycle, need at most two stacks[11]. For a directed acyclic graph, an upward planar drawing is a planar embedding where all the edges are monotonic upward curves. A directed acyclic graph that admits such a drawing is called upward planar. The maximum number of stacks needed for upward planar graphs is unknown but the upper bound was recently reduced to $O(n^{2/3}log^{2/3}(n))$ by Jungeblut et al. where n is the number of vertices in the graph[12]. While Nöllenburg and Pupyrev were able to classify some families of outerplanar directed acyclic graphs with constant stack number, in general it is also unknown if the stack number of outerplanar directed acyclic graphs is bounded by a constant[17].

We now look at the computational complexity of calculating the stack number of graphs.

For maximal planar graphs the stack number is at most 2 if and only if there exists a Hamiltonian cycle. However, finding Hamiltonian cycles in maximal planar graphs is $\mathcal{NP}$-complete[23] and therefore computing the stack number of a graph is also $\mathcal{NP}$-complete.

When the order of vertices is fixed, computing the stack number is still $\mathcal{NP}$-complete. For a fixed order of vertices the problem is equivalent to the circle graph coloring problem. Circle graphs are graphs that are created by looking at a circle with chords. Each chord corresponds to a vertex and two vertices are adjacent if and only if their chords intersect. The circle graph coloring problem searches for a coloring of vertices of circle graphs such that no two adjacent vertices have the same color. For the circle graph coloring problem $\mathcal{NP}$-completeness was proven by Garey et al.[7].

In general, the problem of finding the stack number for directed acyclic graphs is also $\mathcal{NP}$-complete since testing the existence of a 6-stack layout was shown to be $\mathcal{NP}$-complete by Heath and Pemmaraju[10]. However, for 1-stack layouts they gave a linear time algorithm. Binucci et al. were able to prove that determining if a k-stack layout exists is $\mathcal{NP}$-complete for $k \geq 3$[4].

1.3 Contribution

In this thesis, we prove the forest stack numbers and bounds on forest stack numbers as presented in Table 1.1.

Table 1.1: The stack number and forest stack number or tight bounds on the stack number and forest stack number for some graphs.

Graph	**Stack Number**	**Forest Stack Number**
$G = K_n$	$\mathrm{sn}(G) = \lceil n/2 \rceil$ [3]	$\mathrm{sn_f}(G) = \lceil n/2 \rceil$
$G = K_{n,m}$ with $n \ll m$	$\mathrm{sn}(G) = n$ [3]	$\mathrm{sn_f}(G) = n$
G outerplanar	$\mathrm{sn}(G) = 1$	$\mathrm{sn_f}(G) \leq 2$
G planar 3-tree	$\mathrm{sn}(G) \leq 3$ [9]	$\mathrm{sn_f}(G) \leq 3$
G k-tree	$\mathrm{sn}(G) \leq k + 1$ [13]	$\mathrm{sn_f}(G) \leq k + 1$

Since for all graphs, besides outerplanar graphs, the forest stack number or bound on the forest stack number is the same as for the stack number, we provide graphs $F_{n,k}$ with $\mathrm{sn_f}(F_{n,k}) = \mathrm{sn}(F_{n,k}) + 1$ for every $k \in \mathbb{N}$ and n chosen large enough.

We find counterexamples for Conjecture 1.1 if we assume that the order of vertices σ is fixed. For a fixed order of vertices σ and $k \in \mathbb{N}$ we find a graph G with $\mathrm{sn}(G) = k$ and $\mathrm{sn_f}(G)/\mathrm{sn}(G) \geq 3/2$. By making alterations to the graphs $F_{n,k}$ we also find a counterexample to the conjecture if we assume for a specific path in the graph that all path edges are in a single stack. Furthermore, for infinitely many k we find counterexamples to the conjecture for directed acyclic graphs D with $\mathrm{sn}(D) = k$. In particular, we find a counterexample for upward planar graphs. Finally, we suggest further approaches to attack Conjecture 1.1 and propose several open questions.

1.4 Outline

In the next chapter, some basic definitions are introduced that will be used throughout the work. This includes the formal definitions of stack number and forest stack number. Additionally, some basic observations are made about the relation between the stack number and forest stack number.

In the third chapter, we prove bounds on the stack number of various graph families. We start with complete and complete bipartite graphs. Then we consider outerplanar graphs, planar 3-trees and k-trees. In all these cases besides outerplanar graphs the bound found for the forest stack number is the same as that for the stack number. This poses the question if for every stack number k there is a graph that has stack number k but forest stack number $k + 1$. We find such graphs and prove that they need one more stack for their forest stack layout.

In the fourth chapter, we give some counterexamples to Conjecture 1.1 for forest stack layouts with a fixed vertex order. First, an example of a subhamiltonian graph that has forest stack number 4 but stack number 2 with the given order of vertices. Then we present a family of graphs, where for a given order of vertices the forest stack number is at least 1.5 times the stack number. We then try different alterations to the graphs G from the third chapter with $\mathrm{sn_f}(G) = \mathrm{sn}(G) + 1$. Using this method, we find graphs that need two more stacks under different restrictions. Finally, we also consider the conjecture of $\mathrm{sn_f}(G) \leq \mathrm{sn}(G) + 1$ for directed acyclic graphs and give an upward planar graph as counterexample.

2. Preliminaries

This chapter introduces the basic definitions and notations used in the thesis.

The graphs G we consider are undirected graphs with no loops or multiple edges unless stated otherwise. The set of vertices of a graph G is referred to as $V(G)$ and the set of edges as $E(G)$. Since the graphs are undirected, for an edge $\{u, v\} \in E(G)$ we also write $uv \in E(G)$.

A *subgraph* $\hat{G}$ of a graph G has as its vertices a subset of the original vertices $V(\hat{G}) \subseteq V(G)$. The edges $E(\hat{G})$ are also a subset of the original edges $E(\hat{G}) \subseteq E(G)$. The graph $G[A]$ induced by a set of vertices $A \subseteq V(G)$ is the subgraph of G with vertices A and edges $E(G[A]) = \{uv \in E(G) \mid u, v \in A\}$.

The kth power of a graph G is a graph G^k with $V(G) = V(G^k)$. In the graph G^k, all vertices $u, v \in V(G^k)$ are adjacent if and only if in G there is a path from u to v with length at most k.

2.1 Different graph families

In this section, we introduce the graph families used throughout this thesis.

A *complete graph* K_n has n vertices and any two different vertices are connected by an edge. It therefore has the maximum number of edges $E(K_n) = n(n-1)/2$ that a graph with n vertices can have. A clique is a subset of the vertices of a graph, $C \subseteq V(G)$, such that the subgraph induced by C is a complete graph.

For a *complete bipartite graph* $K_{m,n}$ the set of vertices can be partitioned into two sets X, Y with $V(K_{m,n}) = X \cup Y$ and $X \cap Y = \emptyset$. One of the sets contains m vertices and the other n vertices. Therefore $|V(K_{m,n})| = m + n$. Between two vertices u, v with $u, v \in X$ or $u, v \in Y$ there is no edge. Between two vertices in different sets there is always an edge. Hence, a complete bipartite graph has $n \cdot m$ edges.

Planar graphs are graphs that can be embedded in a plane without any edges crossing. Embedding means that the vertices are assigned pairwise different positions on the plane. Every edge $uv \in E(G)$ is represented by a Jordan curve that starts at u and ends at v. Those are the only two vertices that are allowed to be on the curve. If all vertices and edges are embedded and no two edges cross, it is called a *planar embedding* of the graph. The areas created by cutting along the curves of the edges are called *faces*.

An *outerplanar graph* is a graph that is planar and has a planar embedding such that all vertices are adjacent to one face.

Directed acyclic graphs are directed graphs that contain no directed cycles. For a directed acyclic graph an *upward planar drawing* is a planar embedding of the graph in a plane. Additionally, however, all curves representing the edges of the graph in the planar embedding have to be monotonic upward curves. That is, in the coordinate system of the plane with an x-axis and a y-axis the curves have to be monotonically increasing along one of the axes.

A *k-tree* is a graph that is created by taking a complete graph K_{k+1} and repeatedly adding new vertices v and edges connecting v to k vertices of the graph. The k vertices adjacent to v have to be a clique in the graph.

A *Hamiltonian cycle* of a graph is a cycle in the graph that visits each vertex exactly once. A *Hamiltonian graph* is a graph that contains a Hamiltonian cycle. Graphs that are subgraphs of planar Hamiltonian graphs are called *subhamiltonian graphs*.

2.2 Stack layouts and forest stack layouts

A *stack layout* $\Gamma = (\sigma, \mathcal{S})$ of a graph consists of an ordering σ of its vertices and a partition $\mathcal{S}$ of the edges into stacks. Two edges on the same stack are not allowed to cross with respect to σ, meaning that for no two edges xy and uv with $\sigma(x) < \sigma(y)$ and $\sigma(u) < \sigma(v)$ in the same stack we have $\sigma(x) < \sigma(u) < \sigma(y) < \sigma(v)$ or $\sigma(u) < \sigma(x) < \sigma(v) < \sigma(y)$.
The *stack number* $\operatorname{sn}(G)$ of a graph G is the minimum number of stacks of any stack layout of G.

In this thesis we consider a variation of stack layouts and stack numbers:

Definition 2.1. *A* forest stack layout $\Gamma = (\sigma, \mathcal{S})$ *is a stack layout where the graph on each stack is a forest.*

Definition 2.2. *The* forest stack number $\operatorname{sn_f}(G)$ *of a graph G is the minimum number of stacks of any forest stack layout of G.*

Definition 2.3. *A* forest stack layout $\Gamma = (\sigma, \mathcal{S})$ *of an undirected acyclic graph D is a forest stack layout where the order of vertices σ is a topological ordering of D.*

Let $\sigma = v_0, \ldots, v_{n-1}$ be an order of a stack layout of a graph G. There can be at most n edges between vertices $v_i v_j$ with $j = (i+1) \mod n$ and $i \in \{0, \ldots, n-1\}$. We call these edges between adjacent vertices in the order σ *short edges* of the stack or forest stack layout. While not as intuitive the edge $v_{n-1}v_0$ is also considered a short edge. In any stack layout, short edges do not cross any edges.

The stack number and forest stack number of a graph are defined as the minimum number of stacks needed to find a corresponding layout of that graph. With this definition any order σ of the vertices can be chosen for the layout. We now define the stack and forest stack number of a graph for a fixed order of vertices σ:

Definition 2.4. *The stack number* $\operatorname{sn}(G, \sigma)$ *of a graph G for a fixed order of vertices σ is the minimum number of stacks of any stack layout* $\Gamma = (\sigma, \mathcal{S})$ *of G that uses the given order of vertices σ.*

Definition 2.5. *The forest stack number* $\operatorname{sn_f}(G, \sigma)$ *of a graph G for a fixed order of vertices σ is the minimum number of stacks of any forest stack layout* $\Gamma = (\sigma, \mathcal{S})$ *of G that uses the given order of vertices σ.*

2.3 Basic observations about stack numbers and forest stack numbers

First, we observe that the forest stack number is always larger than or equal to the stack number of a graph.

Lemma 2.6. *For any graph G we have* $\mathrm{sn}(G) \leq \mathrm{sn}_\mathrm{f}(G)$.

Proof. Let G be a graph with $\mathrm{sn}_\mathrm{f}(G) = k$. By definition, there exists a forest stack layout $(\sigma, \mathcal{S})$ using k stacks. Because $(\sigma, \mathcal{S})$ is in particular a stack layout it follows that G $\mathrm{sn}(G) \leq k = \mathrm{sn}_\mathrm{f}(G)$. □

Next, we observe that circular shifts of the order of vertices σ in a stack or forest stack layout do not create crossing edges or cycles. Hence, when constructing stack layouts or forest stack layouts of graphs, the vertices can also be arranged on a circle with the edges inside the circle.

Lemma 2.7. *Circular shifts of the order of vertices σ in a stack or forest stack layout create a new stack layout or forest stack layout using the same stacks.*

Proof. Let $\Gamma = (\sigma, \mathcal{S})$ be a forest stack layout or stack layout of G with $\sigma = v_1, \dots, v_n$. Let $\sigma' = v_2, \dots, v_n, v_1$ be the new order of vertices. Assuming that two edges $uv, xy \in E(G)$ do not cross with regard to σ. Without loss of generality, either $\sigma(u) > \sigma(v) > \sigma(x) > \sigma(y)$ or $\sigma(u) > \sigma(x) > \sigma(y) > \sigma(v)$. If $u \neq v_1$ this order is the same for σ' in either case and the edges still do not cross. If $u = v_1$ the vertices are either sorted $\sigma'(v) > \sigma'(x) > \sigma'(y) > \sigma'(u)$ or $\sigma'(x) > \sigma'(y) > \sigma'(v) > \sigma'(u)$ in σ'. Therefore, the edges do not cross with regard to σ'. Changing the order of vertices also creates no new cycles. Therefore, $\Gamma = (\sigma', \mathcal{S})$ is still a stack layout or forest stack layout, respectively. Repeatedly moving the first vertex to the end can create all circular shifts of σ and the resulting layouts are thus still stack layouts, respectively forest stack layouts, using the same stacks. □

3. Testing the conjecture for different graph families

In this chapter, we begin testing Conjecture 1.1 by determining the forest stack number for different graphs.

3.1 Complete graphs

For complete graphs K_m the stack number was shown to be $\lceil m/2 \rceil$ by Bernhart and Kainen using the following proofs[3].

Lemma 3.1 ([3]). *For $m \geq 4$, $\mathrm{sn}(K_m) \geq \lceil m/2 \rceil$.*

Proof. Let us assume that a stack layout with k stacks exists for K_m. Then it is well known that the subgraph on each stack is an outerplanar graph and therefore contains at most $2m - 3$ edges. Since the m short edges can be at most in one stack, we have for the numbers of edges e in the graph $e \leq k \cdot (m-3) + m$. For a complete graph K_m with $m \geq 4$ and $e = m \cdot (m-1)$ this resolves to $m/2 \leq k$. Since k can only be an integer, it follows that $\mathrm{sn}(K_m) \geq k \geq \lceil m/2 \rceil$.

□

Theorem 3.2 ([3]). *For $m \geq 4$, $\mathrm{sn}(K_m) = \lceil m/2 \rceil$.*

Proof. The stack number of K_m is at least $\lceil m/2 \rceil$ as can be seen in Lemma 3.1. It remains to show that $\mathrm{sn}(K_m) \leq \lceil m/2 \rceil$. If m is odd, K_m is a subgraph of K_{m+1} and therefore $\mathrm{sn_f}(K_m) \leq \mathrm{sn_f}(K_{m+1})$. Since $\lceil m/2 \rceil = \lceil (m+1)/2 \rceil$ for m odd the result for K_m follows from K_{m+1}. Thus, we will only consider the case of $m = 2k$ for some integer k. It can be seen that for a complete graph K_m the order of the vertices σ in the layout is not important. Thus, we assume $\sigma = (v_0, \ldots, v_{m-1})$ for $V(K_m) = \{v_0, \ldots, v_{m-1}\}$.

We draw the vertices $V(K_m)$ in a circular layout in this order and get the edges on the first stack of our layout according to Figure 3.1. More precisely $\mathcal{S}_0 = \mathcal{I} \cup \mathcal{E}_s$ with $\mathcal{I} = \{v_i v_j \in E(K_m) \mid j = m - i \vee j = m - i - 1\} \setminus \{v_0 v_{m-1}, v_{k-1} v_k\}$ and $\mathcal{E}_s$ the short edges of order σ. To get the set of edges on the other stacks we rotate the triangulated $2k$-gon in Figure 3.1 k-times meaning that in the previous figure we replace v_i with $\pi(v_i)$

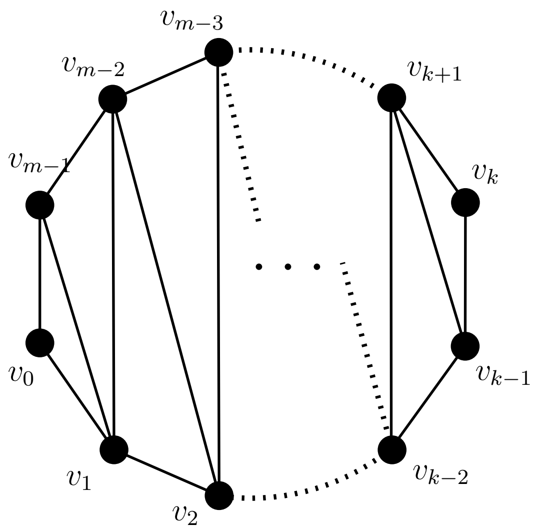

Figure 3.1: First stack in proof of Theorem 3.2.

for $i = 0, \dots, m-1$ where π is defined as $\pi(v_i) = v_{i+1}$ where all indices are taken modulo m. As we have already assigned all short edges to a stack we only put the remaining long edges of the rotated figure on the new stacks. Formally speaking, stack $\mathcal{S}_l$ is defined by

$$\mathcal{S}_l = \{v_{i+l}v_{j+l} \in E(K_m) \mid v_iv_j \in \mathcal{I}\}$$

for $l = 1, \dots, k-1$ where all indices are also taken modulo m.

It can be seen from Figure 3.1 that all edges are assigned only to one stack. We have $2k-3$ long edges per stack and $2k$ short edges on the first stack. Because

$$k(2k-3) + 2k = 2k^2 - k = \frac{m^2}{2} - \frac{m}{2} = \frac{m(m-1)}{2}$$

this accounts for all the edges in the complete graph K_m. Since we can also see from Figure 3.1 that there are no conflicts between the edges on any stack we thus have created a stack layout using $k = m/2$ stacks and therefore $\mathrm{sn}(K_m) = \lceil m/2 \rceil$. □

For complete graphs K_m the forest stack number is also $\lceil m/2 \rceil$. This can be shown by slightly altering the construction used in the proof of the upper bound of the stack number of complete graphs in Theorem 3.2.

Theorem 3.3. *For $m \geq 4$, $\mathrm{sn_f}(K_m) = \lceil m/2 \rceil$.*

Proof. For K_m with $V(K_m) = \{v_0, \dots, v_{m-1}\}$ and $m \geq 4$ Theorem 3.2 and Lemma 2.6 imply that $\mathrm{sn_f}(K_m) \geq \mathrm{sn}(K_m) = \lceil m/2 \rceil$. It remains to show that $\mathrm{sn_f}(K_m) \leq \lceil m/2 \rceil$. For this purpose the edge partition used in the proof of Theorem 3.2 can be slightly modified. The proof of Theorem 3.2 uses the partition of edges created by rotating Figure 3.2 k times. Recall that in Theorem 3.2 the short edges of the graph are all assigned to the first stack. We modify this partition such that two short edges are assigned to each stack according to Figure 3.3. The remaining stacks are created by rotating the first stack. Thus, the stacks $\mathcal{S}_l$ are defined as

$$\mathcal{S}_l = \{v_{i+l}v_{j+l} \in E(K_m) \mid j = m - i \vee j = m - i - 1\}$$

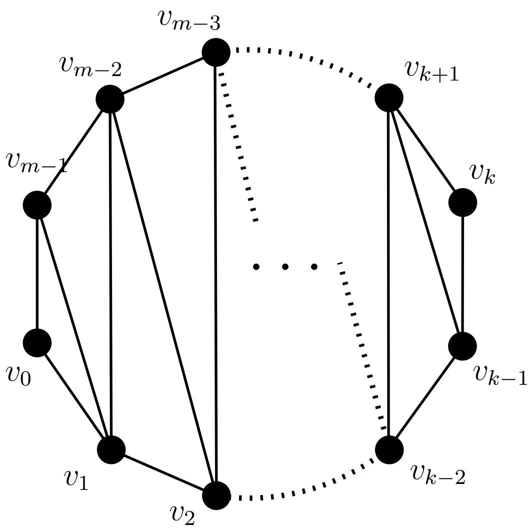

Figure 3.2: First stack in the proof of Theorem 3.2.

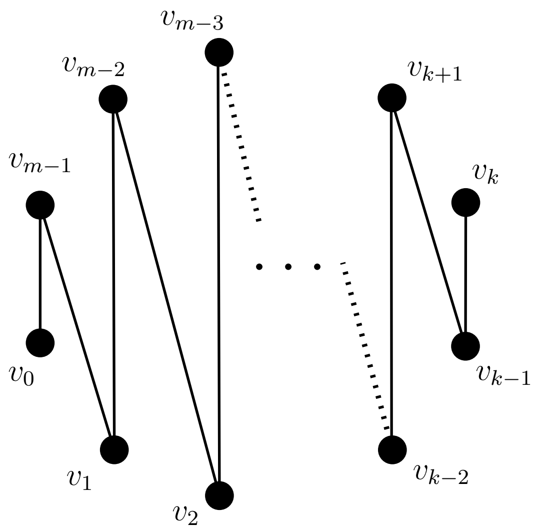

Figure 3.3: First stack in the proof of Theorem 3.3.

for $l = 0, \dots, k-1$. As can be seen in Figure 3.3 no cycles are created and no edges cross. There are $2k-1$ edges in each stack and since

$$k(2k-1) = 2k^2 - k = \frac{m^2}{2} - \frac{m}{2} = \frac{m(m-1)}{2}$$

all edges are assigned a stack. $\square$

3.2 Complete bipartite graphs

For complete bipartite graphs in general the stack number has not been completely determined. In the special case that n is significantly larger than m for a graph $K_{m,n}$ the stack number is m [3]. Under these circumstances, a forest stack layout can be found that uses the same number of stacks.

Theorem 3.4. *For $m \le n$ with $n > m^2 - m + 1$, we have* $\mathrm{sn_f}(K_{m,n}) = m$.

Proof. For m, n given as in Theorem 3.4 the stack number of $K_{m,n}$ is m [3]. Lemma 2.6 implies that $\mathrm{sn_f}(K_{m,n}) \ge \mathrm{sn}(K_{m,n}) = m$. If $K_{m,n}$ has the two sets of vertices $X = \{x_1, \dots, x_m\}$ and $Y = \{y_1, \dots, y_n\}$ we can construct a forest stack layout of $K_{m,n}$ by assigning all edges incident to x_i to a stack $\mathcal{S}_i$ for all i in $1, \dots, m$. Since the edges on each stack form a star, the result is a forest stack layout. Therefore $\mathrm{sn_f}(K_{m,n}) = m$. $\square$

The construction of the forest stack layout in the proof of Theorem 3.4 also shows that $\min(n, m)$ is an upper bound for the forest stack number of complete bipartite graphs $K_{m,n}$ in general. However, this upper bound for the forest stack number is not tight for all complete bipartite graphs. For example, $K_{4,4}$ has a stack layout using three stacks[3] and we can assign the short edges to specific stacks in this stack layout to create a forest stack layout using the same number of stacks.

Theorem 3.5. *The forest stack number of $K_{4,4}$ is* 3.

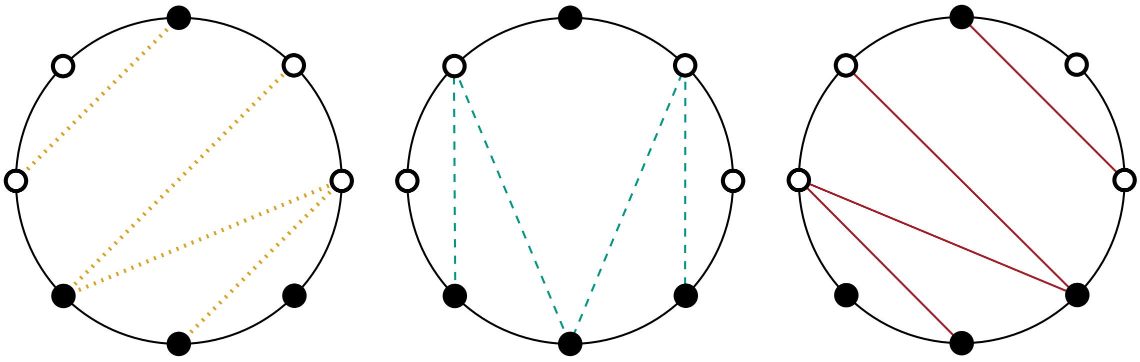

Figure 3.4: 3-Stack layout of $K_{4,4}$ without short edges[3].

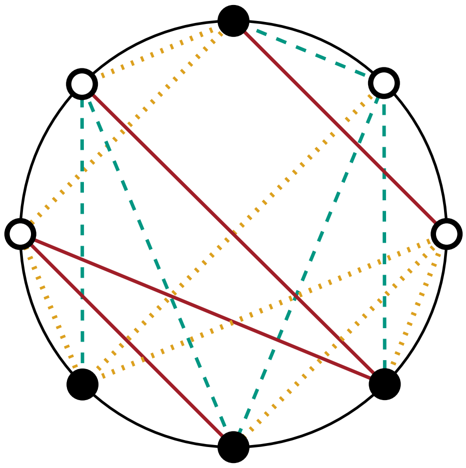

Figure 3.5: Forest 3-stack layout of $K_{4,4}$.

Proof. A stack layout of $K_{4,4}$ is shown in Figure 3.4 where the short edges are not assigned to any specific stack[3]. We assign the short edges to stacks to create the forest stack layout in Figure 3.5. The different line types represent the different stacks. Since the original layout was a stack layout and we only altered the stacks of short edges we have no crossing edges in our layout. Thus, the forest stack number of $K_{4,4}$ is at most 3. Since $|E(K_{4,4})| = 16$ and two forests can contain at most $2|V(K_{4,4})| - 2 = 14$ edges, 3 stacks are necessary. □

It is, however, not the case that for all complete bipartite graphs the stack number equals the forest stack number.

Theorem 3.6. *The stack number of $K_{2,2}$ is 1 but the forest stack number is* 2*.*

Proof. Because $K_{2,2}$ and C_4 are isomorphic, $K_{2,2}$ is an outerplanar graph. Therefore, it has stack number 1[3]. To ensure that all stacks of the layout are forests, we have to move at least one of the edges to a different stack. By moving two incident edges we only have two incident edges per stack and therefore no conflicts in the layout. We have constructed a forest stack layout with two stacks, therefore $\mathrm{sn_f}(K_{2,2}) = 2$. □

The trivial lower bound for $\mathrm{sn}(K_{m,n})$ proved by counting the number of non-short edges per stack is $\lceil (nm - n - m)/(n + m - 3) \rceil$[3]. For the forest stack number we can show the lower bound $\lceil nm/(n + m - 1) \rceil$ because at most $m + n - 1$ edges fit in one forest stack. If the two bounds are considered without rounding up, for $m = n$ the bound for the forest stack number is $(2n^2 - 2n)/(4n^2 - 8n + 3)$ stacks larger than the bound for the stack number. Thus, for large n the bound is improved by about half a stack. Since the forest stack number and stack number can only be integers, in some cases the bound for the forest stack number is one larger than the bound for the stack number. For the case $m = kn$, for some $k \in \mathbb{N}$, further improvements can be made, however never more than one additional stack. For large enough n about $(1 + k^2)/(1 + k^2 + 2k)$ stacks more are needed without rounding.

3.3 Outerplanar graphs

Outerplanar graphs are exactly graphs with stack number 1. We show that the forest stack number of outerplanar graphs is at most 2. Additionally, for any 1-stack layout of an outerplanar graph G we can use the same order of vertices for a forest 2-stack layout of G. We use this fact to construct an upper bound for forest stack numbers in general.

Theorem 3.7. *The forest stack number of an outerplanar graph is at most* 2*. Moreover, the order of vertices in the forest stack layout can be the same as the order of vertices in any given* 1*-stack layout of the graph.*

Proof. Let G be an outerplanar graph with $|V(G)| = n$. It is well known that the stack number of any outerplanar graph is 1[3]. Therefore, there is a stack layout $L = (\sigma, \mathcal{S})$ of G using only one stack. We construct a forest stack layout $\hat{L} = (\sigma, \mathcal{S}_f)$ with two stacks $\mathcal{S}_f = \{\mathcal{S}_1, \mathcal{S}_2\}$. The layout $\hat{L}$ is constructed by retaining the order of the vertices and assigning the edges to $\mathcal{S}_1$ or $\mathcal{S}_2$ such that no stack contains a cycle.

In any outerplanar graph we find a vertex with degree at most two in each subgraph [14]. We will therefore iterate over subgraphs $G_1, \ldots, G_n$ of G. Let v_i be a vertex with degree at most two in G_i for $i = 1, \ldots, n$. We then define $G_n = G$ and $G_i = G_{i+1} - v_{i+1}$ for $i = 1, \ldots, n-1$. In each step we assign all incident edges of v_i in G_i to stacks. We guarantee that after the step in which we consider G_i neither $\mathcal{S}_1$ nor $\mathcal{S}_2$ contain cycles and exactly the edges of G_i are assigned to stacks in the layout. We start with G_1 which contains only one vertex and therefore no edges. Hence, our layout contains the edges of G_1 and has no cycles on any stack. In the step where we consider G_i, v_i has at most degree 2 in G_i. Thus, we assign the incident edges of v_i in G_i such that no stack has more than one of these edges assigned to it. Our inductive assumption guarantees that before this step $\mathcal{S}_1$ and $\mathcal{S}_2$ did not contain any cycles. Hence, the newly assigned edges and therefore also v_1 would have to be part of any cycle that could now exist. However, since v_i has at most degree 1 in each stack, this is not the case. Thus, the stacks $\mathcal{S}_1$ and $\mathcal{S}_2$ do not contain cycles and all edges of G_i are assigned to a stack. Our inductive assumption holds and when we reach G_n after n steps, we have constructed a forest stack layout using only two stacks. The inductive assumption guarantees that each stack of $\hat{L}$ is a forest and since there are no crossings in L and the order of vertices is the same, there are no crossings in $\hat{L}$. It follows that the forest stack number of G is at most 2.

Since this constructive proof of Theorem 3.7 uses the same order of vertices as any given 1-stack layout, the second part of the theorem also holds.

□

Note that Theorem 3.7 is tight since Theorem 3.6 shows that two stacks are necessary for a forest stack layout of $K_{2,2}$.

Corollary 3.8. *For any graph G, $\mathrm{sn_f}(G) \leq 2 \cdot \mathrm{sn}(G)$.*

Proof. Let G be a graph with stack number k. Thus, a stack layout of G exists using k stacks. Since the subgraph on each stack of the stack layout is an outerplanar graph we can create a forest stack layout of the subgraph that uses two stacks while keeping the order of the vertices. Doing so for each stack creates a forest stack layout of graph G using $2 \cdot k$ stacks. □

Corollary 3.8 also implies that for any graph family with bounded stack number, the forest stack number is bounded. For example, planar graphs need at most four stacks for their stack layout [24]. Hence, the forest stack number of planar graphs is at most eight.

3.4 Planar 3-trees

The stack number of planar 3-trees is at most 3[9]. We prove that the stack layout found by Heath in his proof is already a forest stack layout. This can be proven by adding another inductive assumption to his proof. Therefore, we obtain the following Theorem 3.10 that uses Lemma 3.9 in the proof.

Lemma 3.9. *Let x, y, z be three vertices in a stack layout $\Gamma = (\sigma, \mathcal{S})$ where x is directly next to y in the order σ of vertices. Let there be only one edge xz that starts at x on a stack S_i. It follows that $S_i \cup yz$ is also a stack.*

Proof. Let x, y, z be given as in the lemma. Assuming $uv \in S_i$ is an edge that conflicts with edge yz. Then without loss of generality $\sigma(u) > \sigma(y) > \sigma(v) > \sigma(z)$. Since x is next to y this implies $\sigma(u) \geq \sigma(x) \geq \sigma(v) > \sigma(z)$. Since no edge other than xz starts at x it follows that $\sigma(u) > \sigma(x) > \sigma(v)$. Thus, xz is in conflict with uv and therefore any such edge uv cannot exist. □

Theorem 3.10. *The forest stack number of planar 3-trees is at most 3.*

Proof. For a graph G the *stellation* $ST(G)$ is defined by Heath as the graph where in every face of G a new vertex is added with edges connecting to all vertices on the face. Any planar 3-tree is a subgraph of $ST^n(K_3)$ for some n where $ST^n(G) = ST(ST^{n-1}(G))$. Therefore, we prove that a forest stack layout with three stacks exists for $ST^n(K_3)$ for every n. Let the three vertices of K_3 be a, b, c. In $ST(K_3)$ d is added to the interior face and e to the exterior. This graph has a forest stack layout using three stacks which is shown in Figure 3.9. The forest stack layouts for successive stellations of the graph are constructed iteratively. The following invariants are defined by Heath and have to hold for each vertex z that is added into a triangle u, v, w in any step.

(i) For some vertex $x \in \{u, v, w\}$ no other vertices that have been added previously are between x and z in the order of vertices σ.

(ii) The three new edges from z to u, v, w are in three different stacks.

To prove that the layout is also a forest stack layout we add this third invariant:

(iii) The stack layout of the thus far embedded vertices is a forest stack layout.

The basis of Heath's inductive construction is the stack layout of $ST(K_3)$ shown in Figure 3.9. The different types of lines represent the three different stacks. In this layout d is directly next to c and e is next to a. Therefore, the first condition is fulfilled. The second condition also holds as all the edges incident to e and d are on pairwise different stacks. It can also be seen that no stack contains a cycle and therefore it is a forest stack layout. In

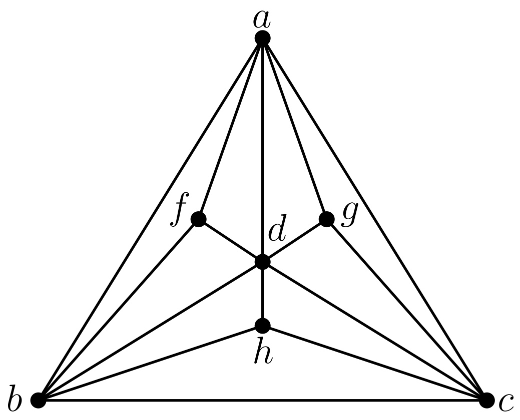

Figure 3.6: $ST^2(K_3)$

each inductive step we add new vertices in the interior of all triangles. We only show how the new vertices added to any $ST(K_3)$ from $ST^{n-1}(K_3)$ are added to the layout. All other vertices are added analogously. In addition, the vertices and edges added to the exterior of the $ST(K_3)$ are also assigned analogously and will therefore not be further mentioned. Thus, we consider three new vertices f, g, h added in each triangle of $ST(K_3)$ around a vertex d to create $ST^2(K_3)$ (see Figure 3.6). The inductive assumption 1 guarantees that for some vertex, say c, there are no previously embedded vertices between c and d. We therefore place f and g between c and d such that $\sigma(d) > \sigma(f) > \sigma(g) > \sigma(c)$. The vertex h is placed directly next to d on the opposite side and we get the order of vertices shown in Figure 3.8. This placement ensures that condition 1 also holds for the newly added vertices. The edges going from the new vertices to d can be added to any of the stacks with no problem since no formerly added vertices are in between. The same is the case for the edge gc. The vertex d has only three adjacent edges. Invariant 1 guarantees that all those edges are assigned to different stacks. The remaining new edges are hc, hb, fb, fa, ga and since the edges dc, db, da exist Lemma 3.9 guarantees that the new edges can be assigned to the stack of the respective assigned edge next to them without causing conflicts. We can thus assign all new edges to stacks as shown in Figure 3.8. Since the stacks of all three edges of each vertex are pairwise different for each vertex f, g, h invariant 2 also holds for the new vertices.

The proof that these two invariants hold is part of Heath's construction of the stack layout. We now have to prove that the third invariant also holds. For this purpose in every inductive step let Γ be the stack layout of $ST^{n-1}(K_3)$ and V_n be the vertices added in this step to create the stack layout Γ_{next}. The second invariant guarantees that for every $v \in V_n$ all three new incident edges are on pairwise different stacks. Hence, the vertices in V_n are not part of any cycle on a stack. Therefore, a possible cycle in Γ_{next} would have to consist only of vertices of $ST^{n-1}(K_3)$. Since no new edges are added between those vertices, this would contradict the third inductive assumption. Thus, Γ_{next} is also a forest stack layout. This inductive step can now be repeated as many times as needed to create a forest 3-stack layout of $ST^n(K_3)$ for any n.

□

3.5 k-Trees

For k-trees there is a tight upper bound of $k + 1$ of their stack number[13]. Similarly to planar 3-trees and complete graphs the proof of this bound constructs a stack layout that is already a forest stack layout.

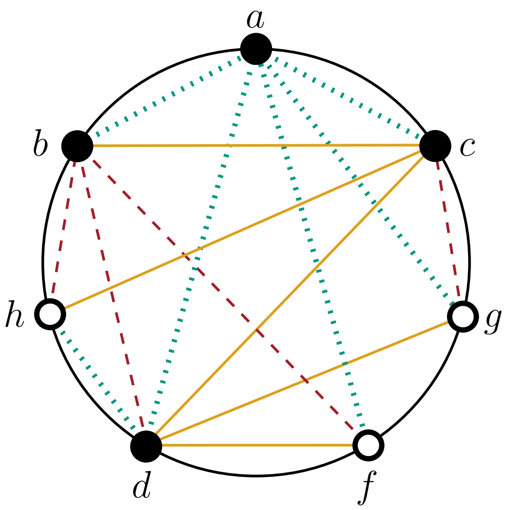

Figure 3.7: Stack layout of $ST^2(K_3)$

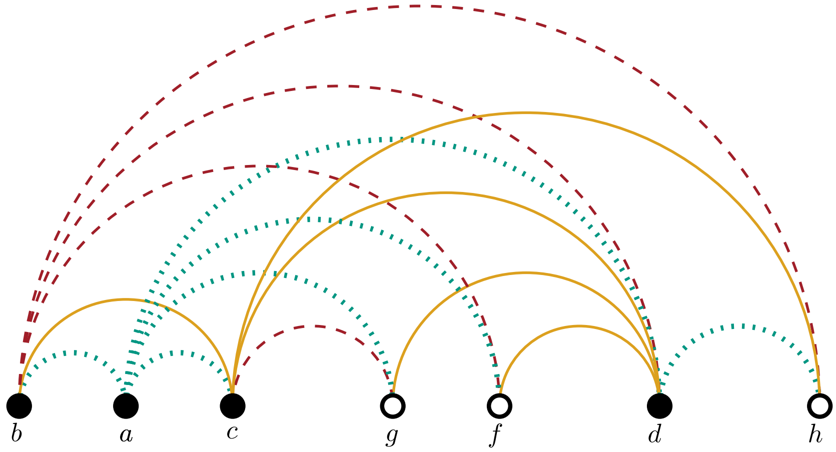

Figure 3.8: Stack layout of $ST^2(K_3)$ in linear layout

In the proof of the stack number of k-trees the following definitions and lemmas are used.

A *tree decomposition* of a graph G is a tuple (X, T) where X is a set of subsets of the vertices $V(G)$ with $X = \{X_1, X_2, \ldots, X_n\}$ and $X_i \subseteq V(G)$. The tree T has as its vertices the sets X_i that are also called bags. Tree decompositions fulfill the following three properties. Firstly, the union of the bags $X_1, \ldots, X_n$ is all vertices of G. Secondly, for every edge $(u, v) \in G$ there has to be a bag X_i with $u, v \in X_i$. And lastly, the subgraph of T induced by all bags containing a vertex u is connected for all $u \in G$.

The *chromatic number* of a graph is the minimum number of colors needed to color the vertices of a graph such that no two adjacent vertices have the same color. For *perfect graphs* the *chromatic number* equals the size of the largest clique in the graph. A *chordal graph* is a graph where there are no induced cycles with more than three vertices. Chordal graphs are an example of perfect graphs.

The following lemma is equivalent to the Helly property:

Lemma 3.11 ([13]). *Let I be an intersection graph of subtrees $T_1, \ldots, T_n$ in a tree T. Then if we have a clique of size k in I there exists a vertex $v \in V(T)$ that is part of at least k subtrees T_i.*

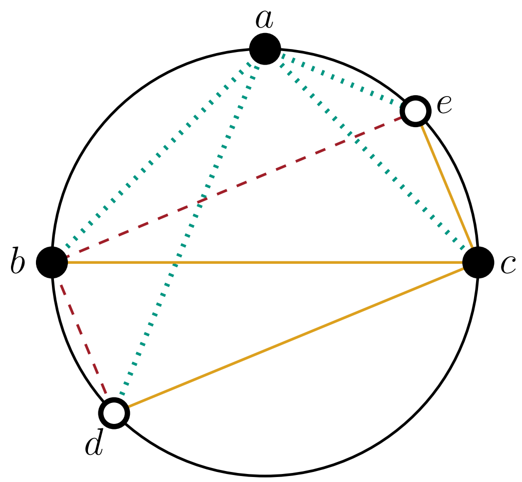

Figure 3.9: Stack layout of $ST(K_3)$

Proof. Assuming a clique C of size k in I exists but every vertex $v \in V(T)$ is part of at most $k-1$ subtrees T_i. Let T_1 be a vertex of I that is part of C. Since no vertex in the subtree T_1 is part of k subtrees we find two other subtrees $T_2, T_3 \in C$ which intersect T_1 at vertices v_2 and v_3 respectively but $v_2 \notin V(T_3)$ and $v_3 \notin V(T_2)$. The subtrees T_2 and T_3 also intersect at a vertex $v_{2,3}$ because they are part of C. This, however, induces a cycle that includes the vertices $v_2, v_3, v_{2,3}$ in T, which is a contradiction. Therefore, there is a vertex $v \in V(T)$ that is part of at least k subtrees. □

Theorem 3.12. *The forest stack number of a k-tree is at most $k+1$.*

Proof. Let us first look at Ganley's and Lenwood's construction of a stack layout on $k+1$ stacks. For any k-tree G it is well known that there is a tree decomposition $D = (X, T)$ of G of width k. To get the order of vertices σ of their stack layout $\Gamma = (\sigma, \mathcal{S})$ they first do a depth-first search of T to get a pre-order $\hat{\sigma}$ of the $X_i \in X$. The order σ is then determined by when an X_i in $\hat{\sigma}$ is first considered with $v \in X_i$. For any $v \in V(G)$ let T_v be the subtree of T induced by all $X_i \in X$ with $v \in X_i$. Since D is a tree decomposition, T_v is connected for all $v \in V(G)$. The intersection graph I of the subtrees T_v of T is a chordal graph[8]. Therefore, I is perfect and the size of the largest clique $\omega(I)$ equals the chromatic number $\chi(I)$. Since D has width k, for any bag X_i of the tree decomposition we have $|X_i| \leq k+1$. If any cliques of size greater than $k+1$ exist in I, Lemma 3.11 implies that there is a bag $X_i \in X$ with more than $k+1$ vertices. This would directly contradict $|X_i| \leq k+1$. Therefore, it follows that cliques in I have at most size $k+1$. Since I is perfect we thus find a coloring using $k+1$ colors. This coloring will be used to color the edges of G. For an edge $uv \in E(G)$ the color $c(uv)$ is defined as follows:

$$c(uv) = \begin{cases} c(T_u) & \sigma(u) < \sigma(v) \\ c(T_v) & \sigma(v) < \sigma(u). \end{cases}$$

Ganley and Lenwood show that no two edges of the same color cross with regard to σ. Let $ab, cd \in E(G)$ with $c(ab) = c(cd)$. If they cross, it can be assumed without loss of generality that $\sigma(a) < \sigma(c) < \sigma(b) < \sigma(d)$. We show that the trees T_a and T_c intersect and therefore ab and cd are not assigned the same color. Let $v_x = X_i$ be the first vertex of the tree decomposition found during the depth-first search with $x \in X_i$. The assumed order σ implies that the vertex v_a containing a was found first in our depth-first search. The edge ab implies that v_b is part of the subtree with root v_a. Since the vertex v_c first containing c is found between the two it is also part of the same subtree. If v_c is on the path from

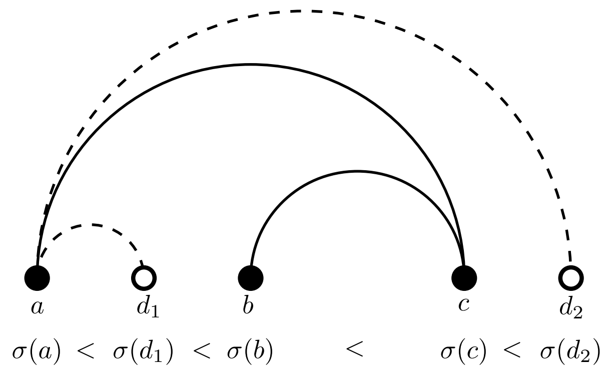

Figure 3.10: Vertices a, b, c in potential cycle in the k-tree stack layout.

v_a to v_b the two trees T_a and T_c intersect. Thus, we assume that v_a, v_b, v_c must be in the configuration C shown in Figure 3.11. The edge cd implies that v_d is part of the subtree with root v_c. However, since v_c is found before v_b in the depth-first search the subtree with root v_c, including v_d, is searched before v_b. This contradicts the assumed order σ found by a depth-first search and therefore the vertices of the tree decomposition cannot be in constellation C. Thus T_a and T_c intersect and ab and cd do not have the same color. The stack layout Γ with the ordering σ and stacks of each individual color of edges therefore is a stack layout of G with $k + 1$ stacks.

We now prove that this stack layout is a forest stack layout. For this purpose, we consider some stack S of the layout and proof that it contains no cycles. First we claim that no edges $ab, bc \in S$ with $\sigma(a) < \sigma(b) < \sigma(c)$ can exists since the edge ab implies that T_a and T_b intersect and therefore $c(ab) = c(T_a) \neq c(T_b) = c(bc)$. Hence, for every cycle $c_1, \ldots, c_k$ in S, we have without loss of generality $\sigma(c_i) < \sigma(c_{i+1})$ for i even and $\sigma(c_i) > \sigma(c_{i+1})$ for i odd. Again, without loss of generality we can thus assume that any cycle contains two edges ac, bc with $\sigma(a) < \sigma(b) < \sigma(c)$ as seen in Figure 3.10. Let ad also be part of the cycle, then $\sigma(a) < \sigma(d)$. We cannot close the cycle with $d = b$ because the edges ab, bc cannot both exist as claimed in the first part of this section. Since the edge ad cannot cross bc, the only potential places for d are between a and b (see d_1 in Figure 3.10) and behind c (see d_2 in Figure 3.10). In the case of $d = d_1$ the next edge would be d_1e with e between a and d_1. However, since $\sigma(a) < \sigma(e) < \sigma(d_1) < \sigma(b)$ any path that closes the cycle from d_1 to b crosses the edge ad_1. Therefore, d_1 is an invalid choice for d. The same is true as well for d_2 since any path from d_2 to b needs to cross the edge ac. Therefore, no cycles exist on any stack of the layout and the given layout is a forest $(k + 1)$-stack layout. □

3.6 Graphs with $\mathrm{sn_f}(G) = \mathrm{sn}(G) + 1$

For all previously looked at graphs besides outerplanar graphs, the stack number equaled the forest stack number. This poses the question if for each k there really exists a graph with $\mathrm{sn}(G) = k$ and $\mathrm{sn_f}(G) = k + 1$. We construct the following graphs that can be shown to have this property.

For any n and $k \leq n$ let $F_{n,k}$ be the graph with vertices $V(F_{n,k}) = \{s_1, \ldots, s_k, v_1, \ldots, v_{n-k}\}$. The edges are defined as $E(F_{n,k}) = \{v_i v_j \mid j = i + 1\} \cup \{s_i v \mid v \in V(F_{n,k})\}$. In Figure 3.12 the graph $F_{8,3}$ is shown as an example. These graphs can also be constructed by first considering a path with $n - k$ vertices. Then k vertices are added iteratively all being adjacent to all previously added vertices.

Theorem 3.13. *For any given k a large enough n can be chosen such that $\mathrm{sn}(F_{n,k}) = k$ and $\mathrm{sn_f}(F_{n,k}) = k + 1$.*

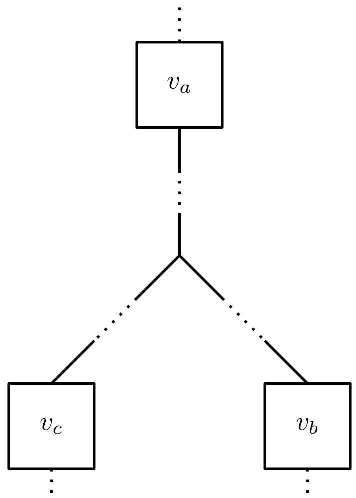

Figure 3.11: Possible structure of v_a, v_b, v_c in the tree decomposition.

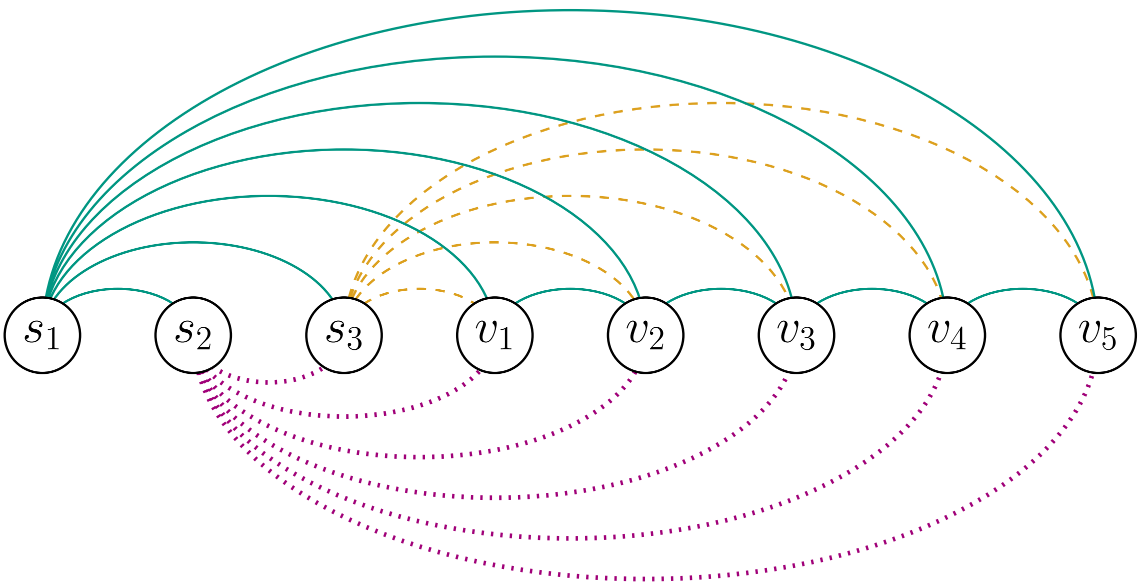

Figure 3.12: The graph $F_{8,3}$ with a 3-stack layout.

Proof. For any n, k we calculate the number $|E(F_{n,k})|$ of edges in $F_{n,k}$. There are $n-k-1$ edges between vertices $v_i v_{i+1}$ with $i = 1, \dots, n-1$. For each s_i there are an additional $n-i$ edges. Thus, it follows that $|E(F_{n,k})| = (n-1)+(n-2)+\cdots+(n-k-1) = (k+1)n-(k+1)(k+2)/2$.

The layout $(\sigma, \mathcal{S})$ with order of vertices $\sigma = s_1, \dots, s_k, v_1, \dots, v_{n-k}$ and the star at s_i for $i \leq k$ on stack $\mathcal{S}_i$ with all remaining short edges also assigned to stack S_1 is a stack layout. Therefore, $F_{n,k}$ has at most stack number k. For a forest stack layout with k stacks the average number of edges per stack are $|E(F_{n,k})|/k = n + n/k - k/2 - 3/2 - 1/k$. If this average is greater than $n-1$ there is a stack with more than $n-1$ edges, which is not possible in a forest stack layout.

$$
\begin{aligned}
n + n/k - k/2 - 3/2 - 1/k &\overset{!}{>} n - 1 \\
\Leftrightarrow \qquad n &\overset{!}{>} k^2/2 + k/2 + 1
\end{aligned}
$$

Therefore, for a large enough n the forest stack number of $F_{n,k}$ is at least $k+1$. The forest stack number is exactly $k+1$ since changing the k-stack layout by putting all edges $v_i v_{i+1}$ on an additional stack creates a forest stack layout with $k+1$ stacks.

We now show that for $n > k^2/2 + k/2 + 1$ a stack layout of $F_{n,k}$ needs k stacks. A stack layout with $k-1$ stack contains at most $(k-1)(n-3)+n$ edges. Therefore, the number of edges should exceed this number so that the stack layout needs at least k stacks:

$$\begin{aligned} (k+1)n - (k+1)(k+2)/2 &\overset{!}{>} (k-1)(n-3)+n \\ \Leftrightarrow \qquad n &\overset{!}{>} k^2/2 - 3k/2 + 4 \end{aligned}$$

For $k \geq 2$ the chosen $n > k^2/2 + k/2 + 1$ fulfills this condition and thus exactly k stacks are needed for a stack layout of $F_{n,k}$.

□

3.7 Subhamiltonian Graphs

Since the stack number of subhamiltonian graphs is 2, Conjecture 1.1 suggests that the forest stack number might be at most 3. We can find a subhamiltonian graph with forest stack number 3. Note that $F_{n,2}$ is subhamiltonian for all n. Additionally, Theorem 3.13 guarantees that we can chose an n such that $\mathrm{sn_f}(F_{n,2}) = 3$. We also know that $\mathrm{sn_f}(G) \leq 4$ for subhamiltonian graphs G because of Corollary 3.8. Thus, there might be a subhamiltonian graph G with $\mathrm{sn_f}(G) = 4$. Finding such a graph would disprove Conjecture 1.1.

Stack layouts of subhamiltonian graphs have an additional property. The order of vertices in a 2-stack layout of a subhamiltonian graph corresponds to a Hamiltonian cycle if all missing short edges are added to the graph. Reversely, any Hamiltonian cycle or order of vertices that can be made into a Hamiltonian cycle by adding edges, such that the graph remains planar, can be chosen as an ordering for the vertices in the 2-stack layout. The graph given in Figure 4.2 and Theorem 4.1 prove that this is not the case for forest stack layouts. Thus, there is a subhamiltonian graph G with a Hamiltonian cycle such that using the Hamiltonian cycle as the order σ of vertices, we have $\mathrm{sn}(G,\sigma) = 2$ and $\mathrm{sn_f}(G,\sigma) = 4$.

4. Counterexamples

4.1 Counterexamples with fixed order of vertices

For stack layouts and forest stack layouts when using a fixed order σ of vertices we can find counterexamples to Conjecture 1.1. Additionally, for any $n \in \mathbb{N}$ we can find a graph G with a fixed order σ of vertices, such that $\mathrm{sn_f}(G, \sigma) - \mathrm{sn}(G, \sigma) > n$.

First, we show that a subhamiltonian graph exists that, using a fixed order σ of vertices, needs at least four stacks for a forest stack layout.

Theorem 4.1. *The graph G in Figure 4.1 with the order σ of vertices in Figure 4.1 has* $\mathrm{sn}(G, \sigma) \leq 2$ *and* $\mathrm{sn_f}(G, \sigma) \geq 4$.

Proof. Consider the graph G with vertex order σ in Figure 4.1. Note that $\mathrm{sn}(G, \sigma) \leq 2$. Consider a 3-forest stack layout where the edges on the three stacks are colored orange, red and green respectively. In Figure 4.2 the graph is shown with all edges colored as follows. To begin with, we consider the central triangle h, j, l. Since no cycles are allowed on a stack, at least two edges have different colors. Without loss of generality, let hj be orange and hl red. It follows that all edges incident to i are green since they cross hj and hl. Thus, eg has to be either orange or red since it would close a circle on the green stack. Because the remaining edges that we consider have no interactions with any edges that are already orange or red, we can without loss of generality let eg be red. Hence, the edges incident to f are orange since they cross the red and green edge incident to e. The edge bd cannot be orange since that would close a cycle on the orange stack. It can also not be green as it crosses the green edge ci. Therefore, bd must be red. It is now impossible for the edge ac to be any of the three colors. It is not red or orange since it crosses the edges incident to b. Furthermore, it is also not green since that would close a cycle on the green stack. Therefore, we have shown that using the order of vertices σ more than three stacks are needed for a forest stack layout. □

If all short edges are added to the graph in Figure 4.1 it would still be a subhamiltonian graph, since adding short edges does not increase the stack number. After adding the short edges, the fixed order σ of vertices in Figure 4.1 is a Hamiltonian cycle. However, because of Theorem 4.1 the graph still needs at least four stacks for a forest stack layout using the order of vertices σ.

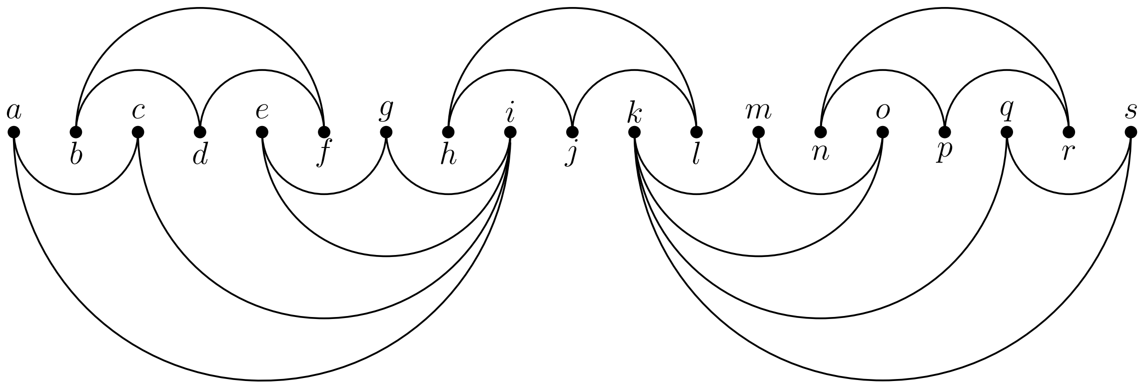

Figure 4.1: Graph that for shown order of vertices needs 4 stacks for forest stack layout.

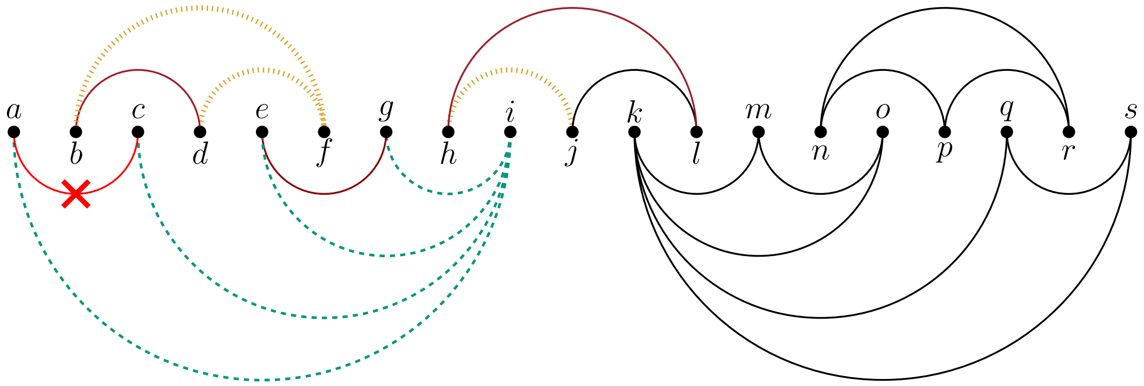

Figure 4.2: Trying to use only 3 stacks for a forest stack layout.

Next, we introduce graphs T_k with order σ of vertices such that the difference between the forest stack number and stack number using the fixed order σ increases linearly in k.

Theorem 4.2. *For infinitely many k, there are G and σ subject to $\operatorname{sn}(G, \sigma) = k$ and $\operatorname{sn_f}(G, \sigma)/\operatorname{sn}(G, \sigma) \geq 3/2$.*

Proof. We define the graph T_k to consist of k triangles K_3. In the i-th triangle K_3^i with $i = 1, \ldots, k$ we name the vertices a_i, b_i and c_i. The order of vertices σ of the graph T_k is set as $\sigma = a_1, \ldots, a_k, b_1, \ldots, b_k, c_1, \ldots, c_k$. As an example, the graph T_3 can be seen in Figure 4.3 with the vertices ordered according to σ. We call all edges a_ib_i the left edges, all edges b_ic_i the right edges and all edges a_ic_i the upper edges. For any $i \neq j$ the edges a_ib_i and a_jb_j cross with regard to σ. Therefore all left edges pairwise cross. It can be seen that the same applies to the right and upper edges respectively. Hence, any stack in a stack layout using σ can contain at most 3 edges: a left edge, a right edge and an upper edge. We now try to pick three such edges. Using the order σ, a left edge a_ib_i does not conflict with a right edge b_jc_j if and only if $i \leq j$. An upper edge a_lc_l does not conflict with the edge a_ib_i if and only if $l \leq i$. Lastly, the upper edge a_lc_l does not conflict with the right edge b_jc_j if and only if $l \geq j$. It follows that only when $i = j = l$ do the edges not conflict. Therefore, three edges fit on one stack only when they are from the same triangle. By assigning all edges in a triangle K_3^i to a stack $\mathcal{S}_i$, a stack layout with three edges on each stack is found. Therefore, the stack number of T_k using order σ is k. However, when constructing a forest stack layout putting three edges from the same triangle on a stack is not allowed. Thus, the forest stack number is at least $3k/2$. The forest stack number is exactly $\lceil 3k/2 \rceil$ because each two triangles can be assigned to three stacks, as shown in Figure 4.4. Hence, $\operatorname{sn_f}(T_k, \sigma)/\operatorname{sn}(T_k, \sigma)) \geq 3/2$ and T_k with order σ fulfills the conditions of the theorem. $\square$

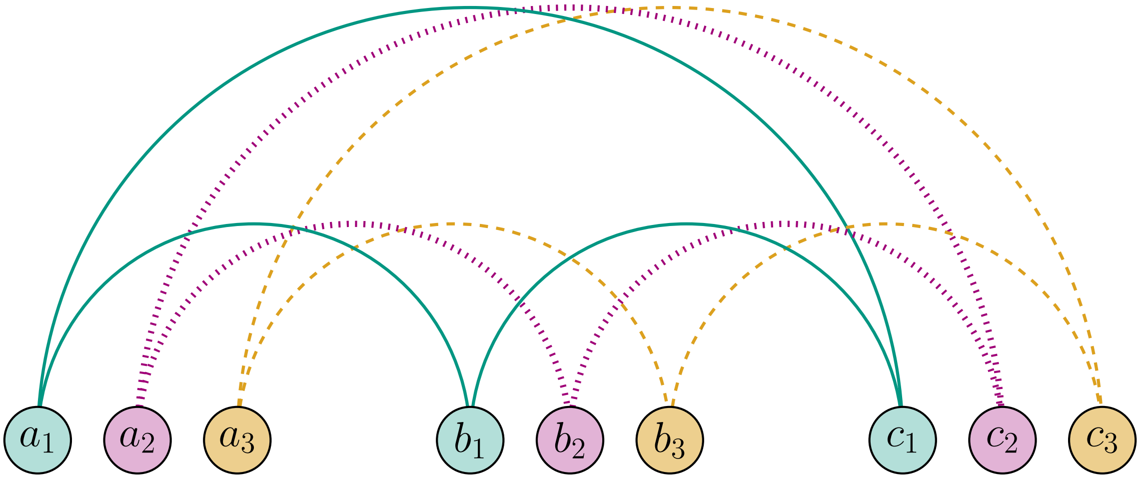

Figure 4.3: Example of T_3 as defined in the proof of Theorem 4.2.

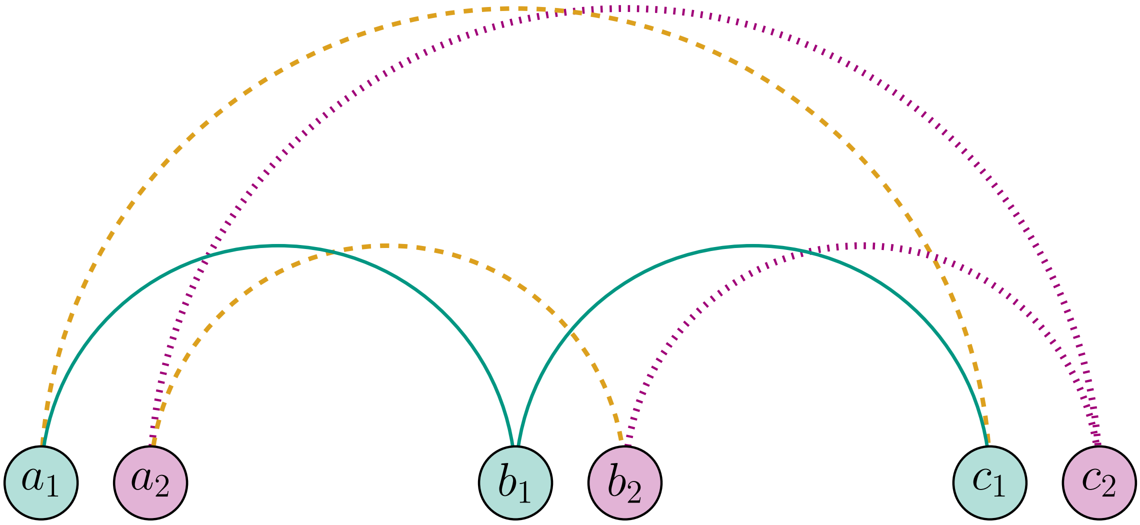

Figure 4.4: T_2 as defined in the proof of Theorem 4.2 with forest stack layout on three stacks.

4.2 Modifications to graphs from Section 3.6

For the graphs G defined in Section 3.6 the number of edges already guarantees that $\mathrm{sn_f}(G) \geq \mathrm{sn}(G) + 1$. We preserve this property and alter these graphs with the goal of increasing the forest stack number by one more. The stack number stays the same and thus a difference of two would be achieved. While we do not find such a graph G with $\mathrm{sn_f}(G) \geq \mathrm{sn}(G) + 2$, if we assume that some more restrictions apply, we find graphs that need two more stacks for a forest stack layout than a stack layout.

We first consider a simple modification that we use in various ways. Recall that graph $F_{n,k}$ consists of a path with $n - k$ vertices named $v_1, \ldots, v_{n-k}$ from one end of the path to the other. We call the vertices on the path *path vertices* and the edges between path vertices $v_i v_{i+1}$ for $i \in \{1, \ldots, n-k-1\}$ *path edges*. There are an additional k vertices in $F_{n,k}$ named $s_1, \ldots, s_k$ that form a clique and also are each adjacent to all path vertices. We call the vertices in this clique *non-path vertices*. A formal definition of $F_{n,k}$ can be found in Section 3.6. We now define a switching operation $\mathrm{sw}_{i,j}$ on graphs $F_{n,k}$ that removes one edge from the graph but adds a different new edge. For $1 \leq j \leq k$ and $2 \leq i \leq n-k-1$ the graph $\mathrm{sw}_{i,j}(F_{n,k})$ is the graph $(V(F_{n,k}), E)$ with the edges $E = (E(F_{n,k}) \setminus s_j v_i) \cup v_{i-1} v_{i+1}$.

Hence, in the new graph s_j and v_i are not adjacent but the new edge $v_{i-1}v_{i+1}$ is added. As an example, the graph $\mathrm{sw}_{4,1}(F_{8,3})$ is shown in Figure 4.5.

It can be observed that the switching operation does not change the stack number and therefore $\mathrm{sn}(\mathrm{sw}_{i,j}(F_{n,k})) = \mathrm{sn}(F_{n,k}) = k$ for $1 \leq j \leq k$ and $2 \leq i \leq n-k-1$ and n chosen large enough so that $\mathrm{sn}(F_{n,k}) = k$ and $\mathrm{sn_f}(F_{n,k}) = k+1$ according to Theorem 3.13. The stack number does not increase since a k-stack layout Γ of $F_{n,k}$ is given by placing all edges adjacent to each non-path vertex s_j on a stack S_j such that each of the k stacks contains a star. The remaining path edges are added to the first stack S_1. The order of vertices σ is given by $\sigma = s_1, \ldots, s_k, v_1, \ldots, v_{n-k}$. The switching modification $\mathrm{sw}_{i,j}$ does not increase the stack number, since the new edge $v_{i-1}v_{i+1}$ can be added to the stack S_j in the layout Γ after the edge s_jv_i is removed. Therefore, we have found a k-stack layout of $\mathrm{sw}_{i,j}(F_{n,k})$ and the stack number of the modified graph is at most k. Since the proof that $\mathrm{sn}(F_{n,k}) \geq k$ in Theorem 3.13 only uses the number of edges, which has not changed, it follows that $\mathrm{sn}(\mathrm{sw}_{i,j}(F_{n,k})) = \mathrm{sn}(F_{n,k}) = k$. As an example the resulting 3-stack layout of the graph $\mathrm{sw}_{4,1}(F_{8,3})$ is shown in Figure 4.5.

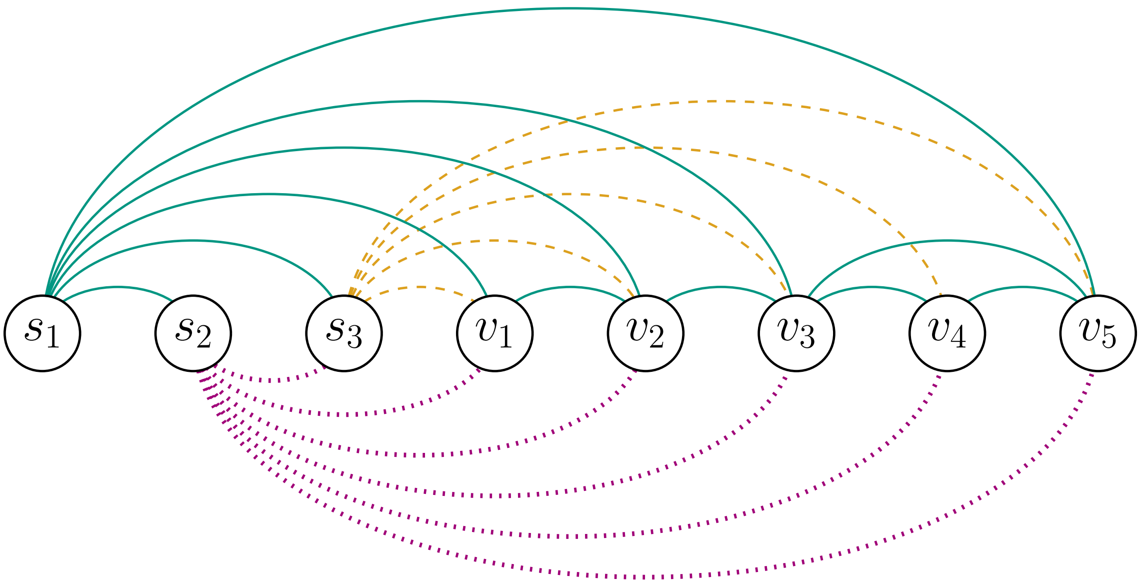

Figure 4.5: The graph $\mathrm{sw}_{4,1}(F_{8,3})$ with a 3-stack layout.

Since the proof of $\mathrm{sn_f}(F_{n,k}) \geq k+1$ in Theorem 3.13 also only uses the number of edges in the graph, which stays the same, the forest stack number after the modification is still at least one greater than the stack number. For the graph $F_{n,k}$ a forest $(k+1)$-stack layout is given by placing all the edges adjacent to a non-path vertex s_j on the same stack S_j, creating stars on these stacks. The path edges are put on an additional stack S_p. We first investigate stack layouts with all path edges in the same stack S_p. Under this assumption, by using the above switching modification the pattern of stars can be broken.

Lemma 4.3. *Let Γ be a forest $(k+1)$-stack layout of the graph $\mathrm{sw}_{i,j}(F_{n,k})$ with $1 \leq j \leq k$ and $2 \leq i \leq n-k-1$ that places all path edges on a single stack S_p. Then there exists an $l \in \{1, \ldots, k\}$ such that the two edges s_lv_{i-1} and s_lv_{i+1} are on different stacks in Γ.*

An example of the effects of Lemma 4.3 for the graph $\mathrm{sw}_{4,1}(F_{8,2})$ is shown in Figure 4.6.

Proof. The edge $v_{i-1}v_{i+1}$ cannot be on the stack S_p of the path edges since that would close a cycle. We now assume that for every $l \in \{1, \ldots, k\}$ the two edges s_lv_{i-1} and s_lv_{i+1} are in the same stack S_l. Since the path edges together with s_lv_{i-1} and s_lv_{i+1} close a cycle, we have $S_l \neq S_p$ for every $l \in \{1, \dots, k\}$. For every l the stack S_l cannot be the one containing $v_{i-1}v_{i+1}$, since that would also create a cycle. For two different $l_1 \neq l_2$ with $l_1, l_2 \in \{1, \dots, k\}$ the stacks S_{l_1} and S_{l_2} cannot be the same, since that would close the cycle $s_{l_1}v_{i-1}s_{l_2}v_{i+1}$. However, since there are only $k + 1 - 2 = k - 1$ stacks left that could be S_l and k different s_l the assumption is wrong and there exists an $l \in \{1, 2, \dots, k\}$ where the two edges s_lv_{i-1} and s_lv_{i+1} are on different stacks. □

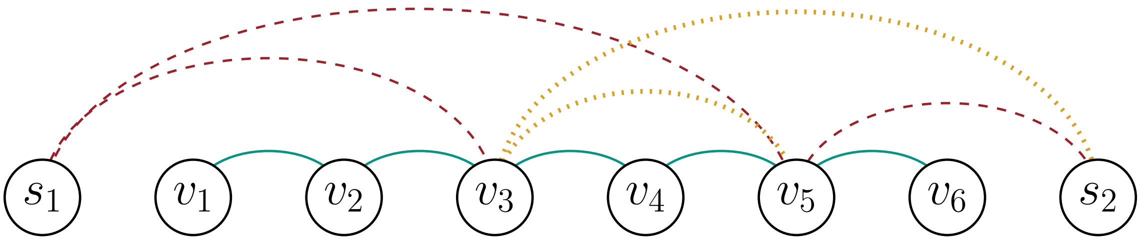

Figure 4.6: A subgraph of $\mathrm{sw}_{4,1}(F_{8,2})$ where the edges s_2v_3 and s_2v_5 are not in the same stack.

The switching operation $\mathrm{sw}_{i,j}$ for $1 \leq j \leq k$ and $2 \leq i \leq n - k - 1$ can be used on a graph $F_{n,k}$ multiple times. If multiple operations were used we define for a set $X = \{(i_1, j_1), (i_2, j_2), \dots, (i_m, j_m)\}$ the operation $\mathrm{sw}_X(F_{n,k}) = \mathrm{sw}_{i_1,j_1}(\mathrm{sw}_{i_2,j_2}(\dots(\mathrm{sw}_{i_m,j_m}(F_{n,k}))))$. We only use multiple switching operations sw_X on a graph $F_{n,k}$ if for each two tuples $(i, j), (i', j') \in X$ the distance $|i - i'| \geq 3$. We call such sets X that keep the minimum distance *feasible sets*. Only for feasible sets X the different modifications do not affect each other and Lemma 4.3 works for each switching operation independently. Additionally, since the modifications do not affect each other and each modification does not increase the stack number it follows that $\mathrm{sn}(\mathrm{sw}_X(F_{n,k})) \leq \mathrm{sn}(F_{n,k}) = k$. By using multiple switching operations on the graph $F_{n,2}$ we get the following theorem:

Theorem 4.4. *For a large enough n there exists a set X such that $\mathrm{sn_f}(\mathrm{sw}_X(F_{n,2})) \geq \mathrm{sn}(\mathrm{sw}_X(F_{n,2})) + 2$ when always placing all path edges on a stack S_p.*

Proof. For a graph $F_{n,2}$ let the $n - 2$ vertices on the path be called $v_1, \dots, v_{n-2}$ from one end of the path to the other. Let s_1 and s_2 be the vertices connected to all other vertices including each other.

Since the switching operations do not increase the stack number, it follows that $\mathrm{sn}(\mathrm{sw}_X(F_{n,2})) \leq \mathrm{sn}(F_{n,2}) = 2$ for any feasible set X. We can also find a 2-stack layout of $\mathrm{sw}_X(F_{n,2})$ where all path edges are in the same stack. For this we take the stack layout of a $F_{n,2}$ with the vertex order $\sigma = s_1, v_1, \dots, v_{n-2}, s_2$ that places all edges adjacent to s_1 and the path edges in a stack S_1 and all other edges in a stack S_2. Now for each switching operation in X that removes an edge s_jv_i and creates an edge $v_{i-1}v_{i+1}$ we assign the new edge to the stack S_j.

Thus, if for some set X four stacks are needed for a forest stack layout with the path edges on a stack S_p we have $\mathrm{sn_f}(\mathrm{sw}_X(F_{n,2})) \geq \mathrm{sn}(\mathrm{sw}_X(F_{n,2})) + 2$ under this assumption. Therefore, we now consider a forest 3-stack layout Γ of $\mathrm{sw}_X(F_{n,2})$ with the path edges on a stack S_p and find a contradiction.

In the forest 3-stack layout Γ all path edges are on the same stack S_p. Let S_{red} and S_{orange} be the other two stacks. We now modify the graph $F_{n,2}$ with switching operations $\mathrm{sw}_{i,j}$ with $1 \leq j \leq 2$ and $2 \leq i \leq n - 3$. For any two operations $\mathrm{sw}_{i,j}, \mathrm{sw}_{i',j'}$ we keep the minimum distance $|i - i'| \geq 3$ and thus the modifications are independent of each other. Thus, Lemma 4.3 guarantees that for each switching operation we have two edges in a star on different non-path stacks or an edge on the path stack S_p connecting to s_1 or s_2.

Next, we show that by using enough switching operations either the star adjacent to s_1 or the star adjacent to s_2 contains two edges on different non-path stacks. For the sake of contradiction, we assume an edge connecting to s_1 or s_2 is placed on the path stack S_p. Without creating cycles this can happen at most once for s_1 and once for s_2. Therefore, if we do switching operations more than two times, each additional operation ensures either the star adjacent to s_1 or the star adjacent to s_2 contains one more edge on the stack S_{red} and one more edge on S_{orange}.

For $n = 47$ we define the feasible set $X = \{(2,1),(5,1),(8,1),(11,1),(14,1),(17,1),(20,1),(23,1),(26,1),(29,1),(32,1),(35,1),(38,1),(41,1),(44,1)\}$. The operation sw_X consists of fifteen independent switching operations. Two of the operations can result in non-path edges in the path stack S_p. However, for any feasible set containing at least thirteen additional operations, one of the two stars contains at least seven pairs of edges with one edge on S_{red} and one edge on S_{orange}. Without loss of generality, let the star adjacent to s_2 be the one containing these pairs of edges. Additionally, Lemma 2.7 guarantees that without loss of generality we can assume that v_1 is the first vertex in the order of vertices σ of the forest stack layout Γ. Each path vertex can now be either before s_2 or after in the order of vertices σ. We now call all vertices with an edge connecting them to s_2 on the stack S_{red} red vertices and all vertices with an edge connecting them to s_2 on the stack S_{orange} orange vertices. Since there are at least seven red and orange vertices each, either there are at least three red and three orange vertices on one side of s_2 or one side has at most two out of seven red vertices and the other at most two out of seven orange vertices.

We first rule out the second case. Without loss of generality, the left side of s_2 has at most two red vertices and the right at most two orange vertices. For each switching operation $\mathrm{sw}_{i,2}$ that forced the edges $v_{i-1}s_2$ and $v_{i+1}s_2$ to be on different non-path stacks one vertex, either v_{i-1} or v_{i+1}, is colored red and the other orange. The edge $v_{i-1}v_{i+1} \notin S_p$ therefore connects one red vertex and one orange vertex for each $(i,2) \in X$. Since at most two red vertices are on one side and at most two orange vertices on the other there can be at most four edges connecting red and orange vertices that do not go from the left side of s_2 to the right. The number of edges connecting a red and an orange vertex from left to right is also limited because of the problem shown in Figure 4.7. Any edge $e_{o,cross}$ on the orange stack S_{orange} going from the left side to the right can only connect to the leftmost orange vertex before s_2 in the order of vertices σ. Else, the edges going from s_2 to orange vertices further away on the left conflict with $e_{o,cross}$. The same is true for edges $e_{r,cross}$ on the red stack S_{red} going from one side to the other that can only connect to the rightmost red vertex after s_2 in the order of vertices σ. Hence, under the assumption that one side of s_2 has at most two red vertices and the other at most two orange vertices at most six pairs of red and orange vertices can be connected by an edge on S_{red} or S_{orange}. An example of an ordering of vertices and six edges connecting red and orange vertices is shown in Figure 4.8. However, since at least seven pairs of edges were affected adjacent to s_2, seven pairs have to be connected. Hence, the assumption must be false and at least three red and three orange vertices are on one side of s_2.

First, we assume three red and three orange vertices are before s_2 in the order of vertices. We call these six vertices $c_1, \ldots, c_6$ with $\sigma(s_1) < \sigma(c_1) < \sigma(c_2) < \cdots < \sigma(c_6) < \sigma(s_2)$. Since there is at least one red vertex and one orange vertex to the left of c_5 and c_6 the edges s_1c_5 and s_1v_6 cannot be on the stacks S_{red} or S_{orange}. However, placing them both on the path stack would close a cycle. The problematic edges in this stack layout are shown in Figure 4.9. Therefore, in this case the assumption that a forest 3-stack layout Γ exists if all path edges are placed on one stack is false.

Therefore, we assume that three red and three orange vertices are after s_2 in the order of vertices. We call these six vertices $c_1, \ldots, c_6$ with $\sigma(s_1) < \sigma(s_2) < \sigma(c_1) < \sigma(c_2) < \cdots <$

$\sigma(c_6)$. Since there is at least one red vertex and one orange vertex to the right of c_1 and c_2 the edges s_1c_1 and s_1c_2 cannot be on the stacks S_{red} or S_{orange}. However, placing them both on the path stack would close a cycle. The problematic edges in this stack layout are shown in Figure 4.10. Therefore, in this case the assumption that a forest 3-stack layout Γ exists if all path edges are placed on one stack is false.

Hence, at least four stacks are needed for a forest stack layout assuming all path edges are on a stack S_p. Therefore, for any set X of at least fifteen independent switching operations it follows that $\mathrm{sn_f}(\mathrm{sw}_X(F_{n,2})) \geq \mathrm{sn}(\mathrm{sw}_X(F_{n,2})) + 2$ when always placing all path edges in a stack S_p.

□

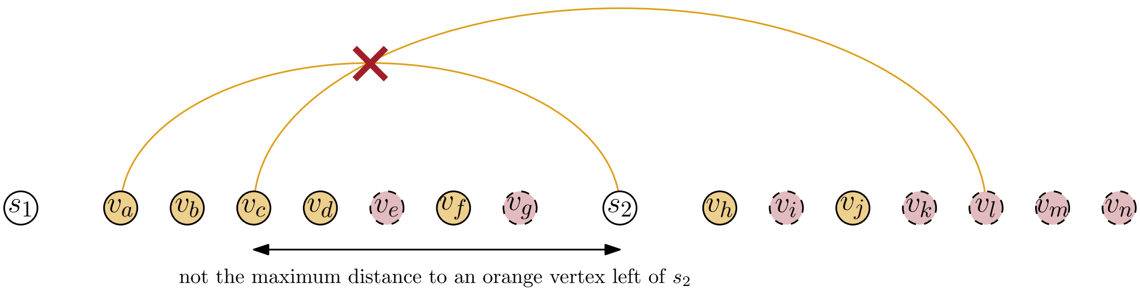

Figure 4.7: Connecting to an orange vertex on the left from the right side that does not have maximum distance from s_2 creates conflicting edges.

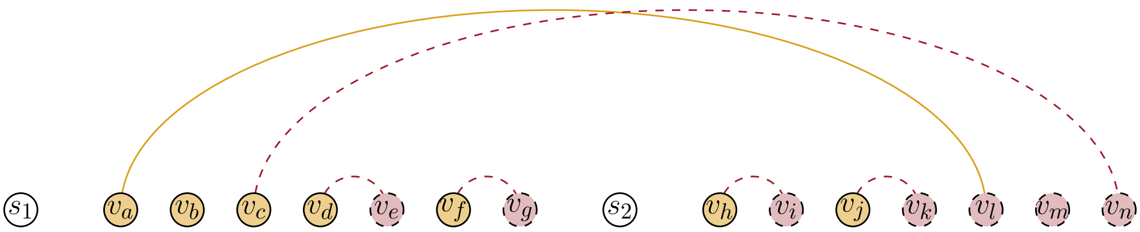

Figure 4.8: An example of the maximum of six possible edges between red and orange vertices with at most two red vertices left of s_2 and at most two orange vertices on the right.

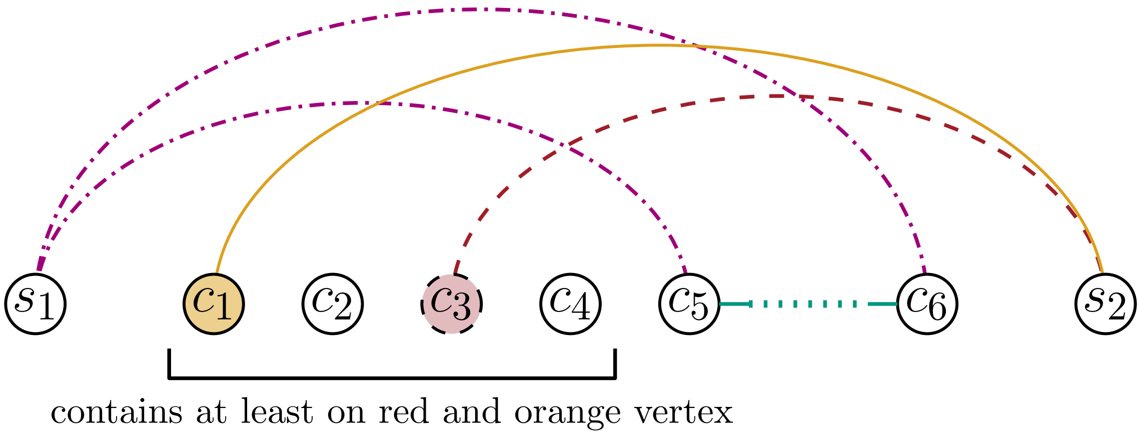

Figure 4.9: Assuming there are three red and three orange vertices to the left of s_2 the purple edges cannot both be placed on the 3 stacks of the stack layout.

In the graphs modified as described above, we always assume that the edges on the path are placed on a single stack S_p. To avoid this constraint we consider the second power

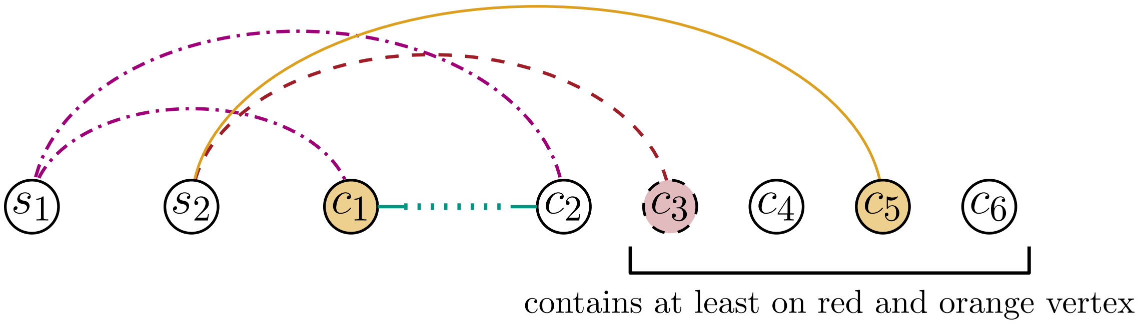

Figure 4.10: Assuming there are three red and three orange vertices to the right of s_2 the purple edges cannot both be placed on the 3 stacks of the stack layout.

of a path. The second power of a path consists of a normal path. Additionally, every vertex v_i in the path is also connected to v_{i+2}. We now construct graphs similar to $F_{n,2}$ but using the second power of paths instead. We call them $P_{n,2}$. Let s_1, s_2 be again two vertices in $V(P_{n,2})$ not part of the second power of a path on $n-2$ vertices. One of these vertices, say s_1 is connected to all even vertices in the second power of the path, and the other to all odd vertices. We get the same graph if we apply the switching operation $\mathrm{sw}_{i,j}$ described above on a $F_{n,2}$ multiple times and remove the edge $s_1 s_2$. In this case, the switching operations do not keep the minimum distance to be independent. For the order of vertices $\sigma = s_1, v_1, v_2, \ldots, v_{n-2}, s_2$ the stack number of $P_{n,2}$ is still 2. We consider the 2-stack layout of $F_{n,2}$ where all edges incident to s_1 and all path edges are on a stack S_1 and all other edges are on a stack S_2. For each switching operation that adds a new edge $v_i v_{i+2}$ the new edge can be placed on the stack S_1 if i is even and on stack S_2 is i is odd. This creates a 2-stack layout using the order of vertices σ of the graph $P_{n,2}$. As an example in Figure 4.11 the 2-stack layout of the graph $P_{8,2}$ is shown.

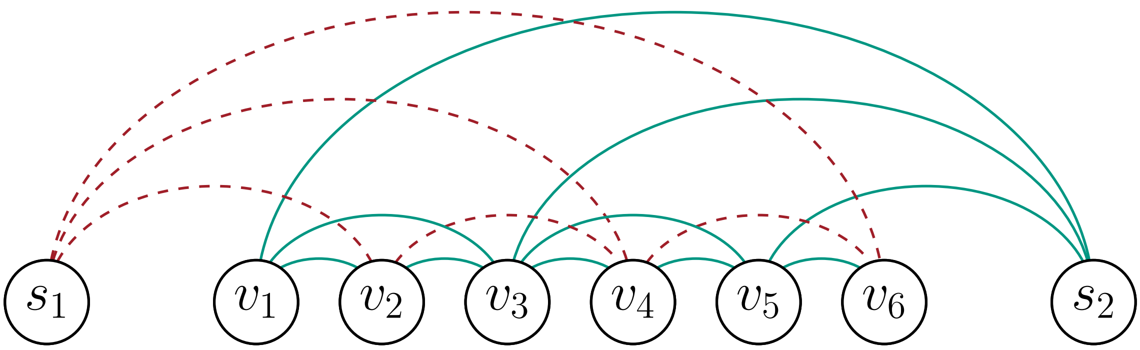

Figure 4.11: A 2-stack layout of the graph $P_{8,2}$.

We now show that the graphs $P_{n,2}$ using the fixed order of vertices σ need at least four stacks for a forest stack layout but only two for a stack layout. Only for a fixed order of vertices the forest stack number is two larger for graphs $P_{n,2}$. However, in contrast to the triangle graphs in Theorem 4.2 the order of vertices σ is optimal for stack layouts of $P_{n,2}$.

Theorem 4.5. *There is a vertex ordering σ such that $\mathrm{sn_f}(P_{n,2}, \sigma) \geq 4$ and $\mathrm{sn}(P_{n,2}, \sigma) = 2$ for $n > 7$.*

Proof. As fixed order of vertices σ we consider the ordering $\sigma = s_1, v_1, v_2, \ldots, v_{n-2}, s_2$. The graph $P_{8,2}$ is shown in Figure 4.12 using the order of vertices σ. Since for $n > 7$ the graph $P_{8,2}$ is a subgraph of $P_{n,2}$ it is sufficient to show that $P_{8,2}$ needs four stacks for a forest stack layout. Let the four edges $v_1v_3, v_2v_4, v_3v_5, v_4v_6$ of $P_{8,2}$ be called e_1, e_2, e_3, e_4 in that order. We try to color the edges of the graph with three colors green(g), red(r) and orange(o). Up to renaming the colors for the four edges $e_1, \ldots, e_4$ because of crossing edges only the following colorings are possible: $(g, r, g, r), (g, r, g, o), (g, r, o, g)$ and (g, r, o, r). For all cases we now find a contradiction. Deriving the color of the edges is always done based only on crossing edges and cycles that would be closed. In the corresponding figures the edges are colored as assumed in the specific case.

Case 1: $c(e_1, e_2, e_3, e_4) = (g, r, g, r)$
The edges e_1, e_2, e_3, e_4 are colored (g, r, g, r) as shown in Figure 4.12. The edge v_3s_2 can only be green or orange. Assuming v_3s_2 is green the following colors follow in the given order: $c(v_5s_2) = o, c(s_1v_6) = r, c(s_1v_4) = o$. Now the edge v_1s_2 cannot be green, red or orange. Therefore, we assume v_3s_2 is orange. In that case the following colors follow in the given order: $c(s_1v_4) = r, c(s_1v_6) = g, c(s_1v_2) = o$. Now the edge v_1s_2 cannot be green, red or orange. Therefore, in the case of $c(e_1, e_2, e_3, e_4) = (g, r, g, r)$ the graph has no forest stack layout on three stacks.

Case 2: $c(e_1, e_2, e_3, e_4) = (g, r, g, o)$
The edges e_1, e_2, e_3, e_4 are colored (g, r, g, o) as shown in Figure 4.13. The edge v_3s_2 can only be green or orange. Assuming v_3s_2 is green the following colors follow in the given order: $c(v_5s_2) = r, c(s_1v_6) = o, c(s_1v_4) = r$. Now the edge v_1s_2 cannot be green, red or orange. Therefore, we assume v_3s_2 is orange. In that case the following colors follow in the given order: $c(s_1v_4) = r, c(s_1v_2) = o, c(v_1s_2) = g, c(v_5s_2) = r$. Now the edge s_1v_6 cannot be green, red or orange. Therefore, in the case of $c(e_1, e_2, e_3, e_4) = (g, r, g, o)$ the graph has no forest stack layout on three stacks.

Case 3: $c(e_1, e_2, e_3, e_4) = (g, r, o, g)$
The edges e_1, e_2, e_3, e_4 are colored (g, r, o, g) as shown in Figure 4.14. The edge v_3s_2 can only be green or orange. Assuming v_3s_2 is green the following colors follow in the given order: $c(s_1v_4) = r, c(s_1v_2) = o$. Now the edge v_1s_2 cannot be green, red or orange. Therefore, we assume v_3s_2 is orange. In that case the following colors follow in the given order: $c(v_5s_2) = r, c(s_1v_6) = g, c(s_1v_4) = r, c(s_1v_2) = o$. Now the edge v_1s_2 cannot be green, red or orange. Therefore, in the case of $c(e_1, e_2, e_3, e_4) = (g, r, o, g)$ the graph has no forest stack layout on three stacks.

Case 4: $c(e_1, e_2, e_3, e_4) = (g, r, o, r)$
The edges e_1, e_2, e_3, e_4 are colored (g, r, o, r) as shown in Figure 4.15. The edge v_3s_2 can only be green or orange. Assuming v_3s_2 is green the following colors follow in the given order: $c(s_1v_4) = r, c(s_1v_2) = o$. Now the edge v_1s_2 cannot be green, red or orange. Therefore, we assume v_3s_2 is orange. In that case the following colors follow in the given order: $c(v_5s_2) = g, c(s_1v_6) = r, c(s_1v_4) = g, c(s_1v_2) = o$. Now the edge v_1s_2 cannot be green, red or orange. Therefore, in the case of $c(e_1, e_2, e_3, e_4) = (g, r, o, r)$ the graph has no forest stack layout on three stacks.

Since all cases lead to a contradiction it follows that $\mathrm{sn_f}(P_{n,2}, \sigma) \geq 4$.

□

For graphs $P_{n,2}$ four stacks are needed for a forest stack layout if the order of vertices is fixed as $\sigma = s_1, v_1, \ldots, v_{n-2}, s_2$. When using a different order σ' of vertices only three forest stacks are needed. Let i be the largest even number such that $v_i \in V(P_{n,2})$ and j be the largest odd number such that $v_j \in V(P_{n,2})$. We define the new order of vertices $\sigma' = v_2, v_4, \ldots, v_i, s_1, s_2, v_j, \ldots, v_3, v_1$.

Lemma 4.6. *The forest stack number of $P_{n,2}$ is 3 for $n > 7$.*

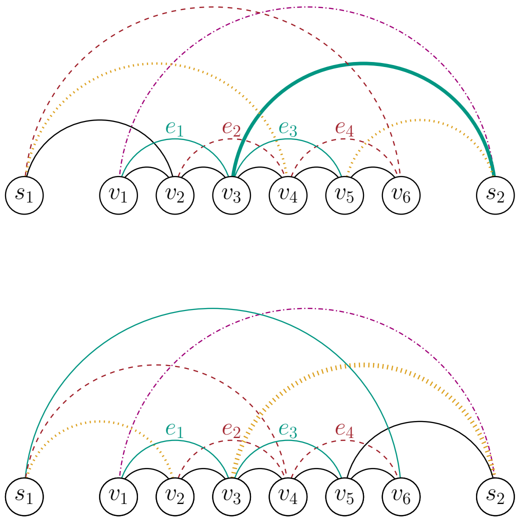

Figure 4.12: The graph $P_{8,2}$ in the case $c(e_1, e_2, e_3, e_4) = (g, r, g, r)$.

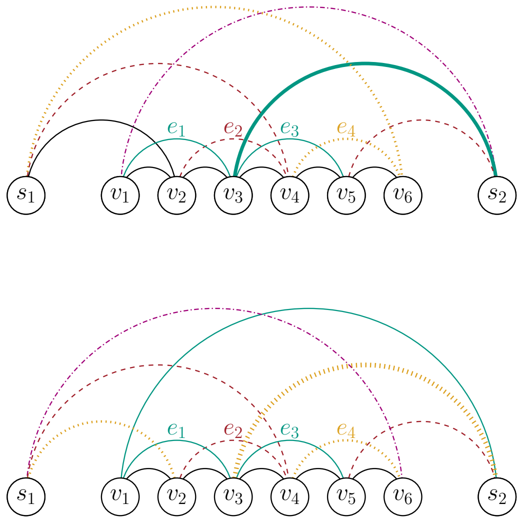

Figure 4.13: The graph $P_{8,2}$ in the case $c(e_1, e_2, e_3, e_4) = (g, r, g, o)$.

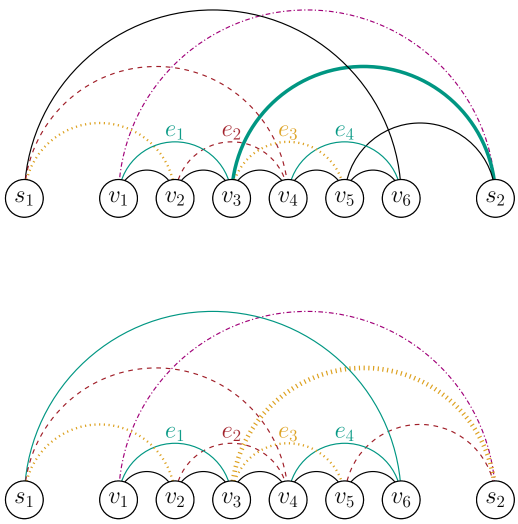

Figure 4.14: The graph $P_{8,2}$ in the case $c(e_1, e_2, e_3, e_4) = (g, r, o, g)$.

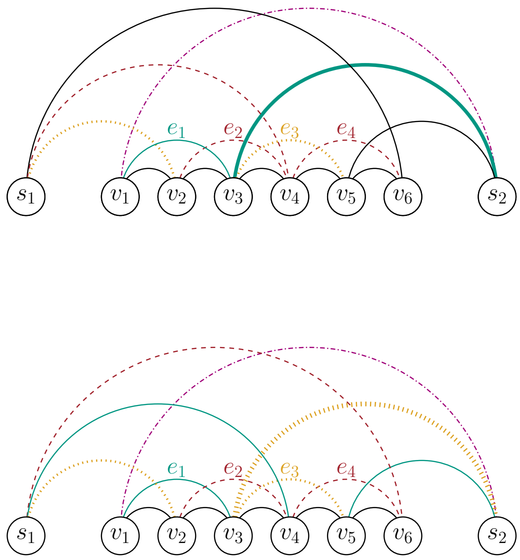

Figure 4.15: The graph $P_{8,2}$ in the case $c(e_1, e_2, e_3, e_4) = (g, r, o, r)$.

Proof. The number of edges in $P_{n,2}$ is $(n-2)+(n-3)+(n-4)=3n-9$. However, a forest 2-stack layout can contain at most $2n-2$ edges. Therefore, for $n>7$ the forest stack number of $P_{n,2}$ is at least 3.

We construct a forest 3-stack layout Γ of $P_{n,2}$ using the order of vertices σ'. As an example the forest 3-stack layout of $P_{8,2}$ is shown in Figure 4.16. All edges adjacent to s_1 are assigned to a stack S_1. The vertex s_1 is adjacent only to vertices v_i with i even, which are before s_1 in the order of vertices σ'. Therefore, at this point in the stack S_1 the vertex s_1 is the rightmost vertex any edge connects to. Since we add a star to S_1 no edges conflict and no cycles are created. The edges between odd vertices in the path are now also added to the stack S_1. Since all odd vertices are behind s_2 in descending order in σ', no cycles are created and no edges cross. Now the edges adjacent to s_2 are added to a stack S_2. Additionally, the edges connecting even vertices on the path are added to S_2. For the same reasons that S_1 contains no cycles and crossing edges, S_2 is also a stack in a forest stack layout. The only remaining edges of $P_{n,2}$ are the edges v_iv_{i+1} on the path. We assign these path edges to a stack S_3. It can be seen that no path edges cross when using the order of vertices σ'. Therefore, we have found a forest 3-stack layout of $P_{n,2}$ using the order of vertices σ'. $\square$

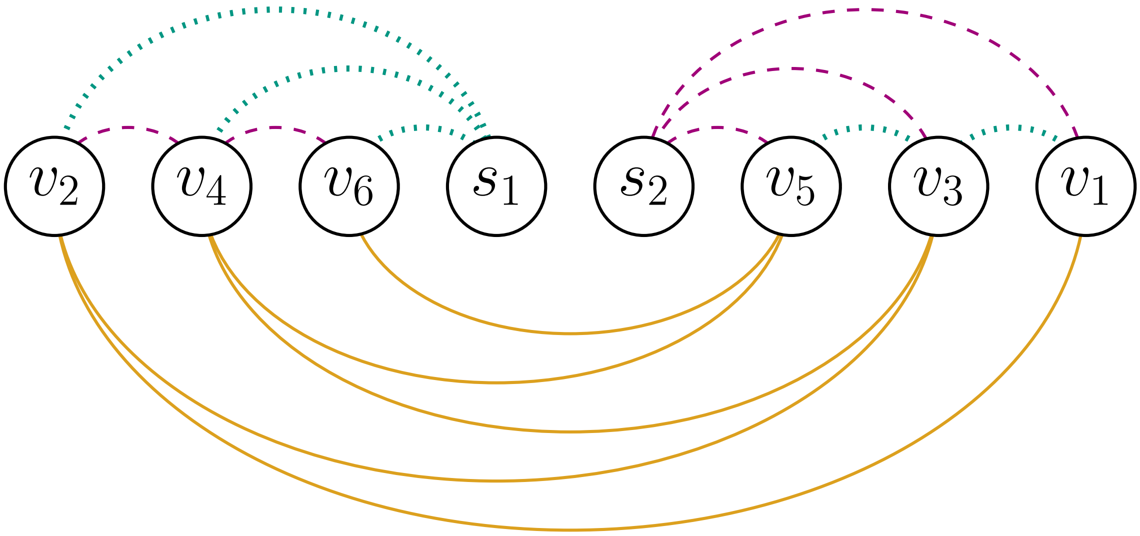

Figure 4.16: Forest 3-stack layout of $P_{8,2}$.

4.3 Counterexamples for directed acyclic graphs

For stack layouts of directed acyclic graphs D, the order of vertices σ is a topological ordering of vertices. Therefore, for two vertices $u, v \in V(D)$ with an edge $uv \in E(D)$ the order is $\sigma(u) < \sigma(v)$. For forest stack layouts of directed acyclic graphs the same restriction applies. Additionally, the corresponding undirected graph on each stack of the forest stack layout has to be a forest. Because the same number of edges can be in a stack of a stack layout of an undirected graph as in a stack for a directed graph the motivation from Section 1.1 still works. Thus, the conjecture $\mathrm{sn_f}(D) \leq \mathrm{sn}(D)+1$ for directed acyclic graphs can be proposed.

However, for directed acyclic graphs a counterexample to this conjecture can be found. In Section 4.1 and Section 4.2 we introduce some graphs G with set orders of vertices σ for which $\mathrm{sn_f}(G,\sigma) \geq \mathrm{sn}(G,\sigma)+2$. We find directed acyclic graphs D with $\mathrm{sn_f}(D) \geq \mathrm{sn}(D)+2$ by altering these graphs. In particular, for the following directed acyclic graphs the difference between the forest stack number and stack number increases linearly with k.

Lemma 4.7. *For infinitely many k, there are directed acyclic graphs D such that* $\operatorname{sn}(D) = k$ *and* $\operatorname{sn_f}(D)/\operatorname{sn}(D) \geq 3/2$.

Proof. We consider the graph T_k from Theorem 4.2 that consists of k triangles K_3. In the i-th triangle K_3^i with $i = 1, \dots, k$ we name the vertices a_i, b_i and c_i. The order of vertices σ of the graph T_k is fixed as $\sigma = a_1, \dots, a_k, b_1, \dots, b_k, c_1, \dots, c_k$. Currently, the graph contains no short edges. We add all short edges and get an undirected graph G. The graph G has stack number k since short edges do not increase the stack number given by Theorem 4.2. We direct all edges of G according to the order σ to get the directed acyclic graph D. As an example, the directed acyclic graph created from T_3 is shown in Figure 4.17. Since D contains all short edges the only topological ordering of vertices in D is σ. Thus, finding a stack layout for D is equivalent to finding a stack layout for G using the fixed order of vertices σ. Since the added short edges can only increase the forest stack number, Theorem 4.2 implies that $\operatorname{sn_f}(D)/\operatorname{sn}(D) \geq 3/2$. □

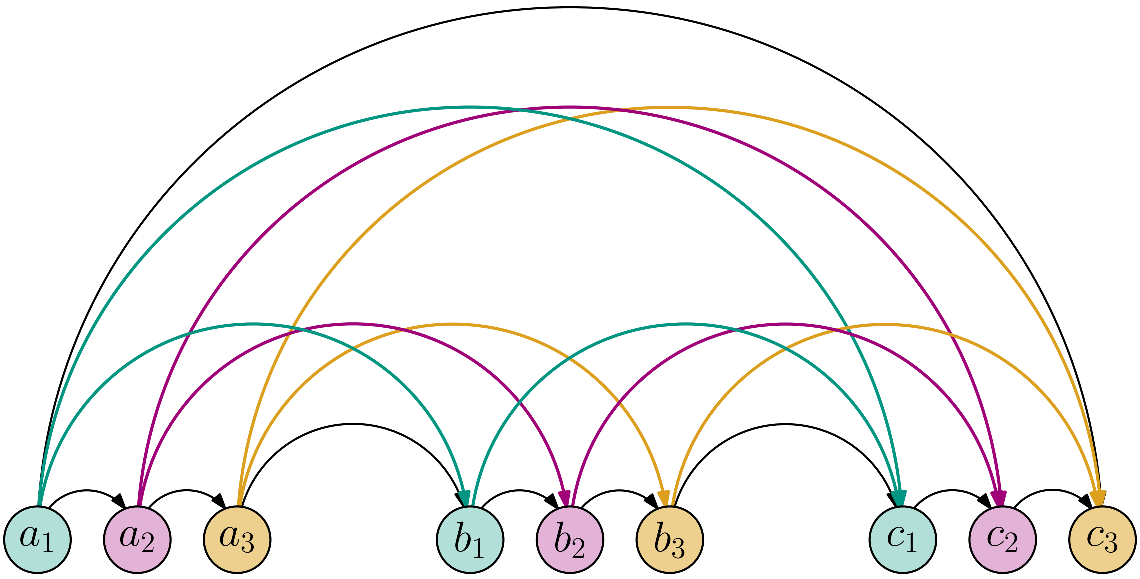

Figure 4.17: The directed acyclic graph created from T_3 in Lemma 4.7.

Lemma 4.7 already gives us a counterexample to Conjecture 1.1 for directed acyclic graphs. However, for upward planar graphs it only proves the existence of an upward planar graph D with $\operatorname{sn}(D) = 2$ and $\operatorname{sn_f}(D) = 3$.

Lemma 4.8. *There is an upward planar graph D for which* $\operatorname{sn}(D) = 2$ *and* $\operatorname{sn_f}(D) \geq 4$.

Proof. We consider the graph $P_{8,2}$ shown in Figure 4.12. We add all missing short edges, that is the edges s_1v_1, v_6s_2 and s_1s_2, to $P_{8,2}$ to get the undirected graph G. Using the order of vertices $\sigma = s_1, v_1, \dots, v_6, s_2$ the graph G still has stack number 2 proven by the stack layout shown in Figure 4.18. We direct all edges of G according to the order σ to get the directed graph D shown in Figure 4.19. Figure 4.19 also proves that an upward planar drawing of D exists and it is therefore an upward planar graph. Since the only topological ordering of vertices in D is σ, finding a stack layout for D is equivalent to finding a stack layout for G using the fixed order of vertices σ. Therefore $\operatorname{sn}(D) = 2$. Finding a forest stack layout of D is also equivalent to finding a forest stack layout of G using the fixed order of vertices σ. Since the added short edges can only increase the forest stack number, Theorem 4.5 implies that $\operatorname{sn_f}(D) \geq 4$. □

The graph $P_{8,2}$ used in Lemma 4.8 is not the only one that can be altered to be an upward planar graph D with $\mathrm{sn}(D) = 2$ and $\mathrm{sn_f}(D) \geq 4$. Equivalently the graph given in Theorem 4.1 can also be used.

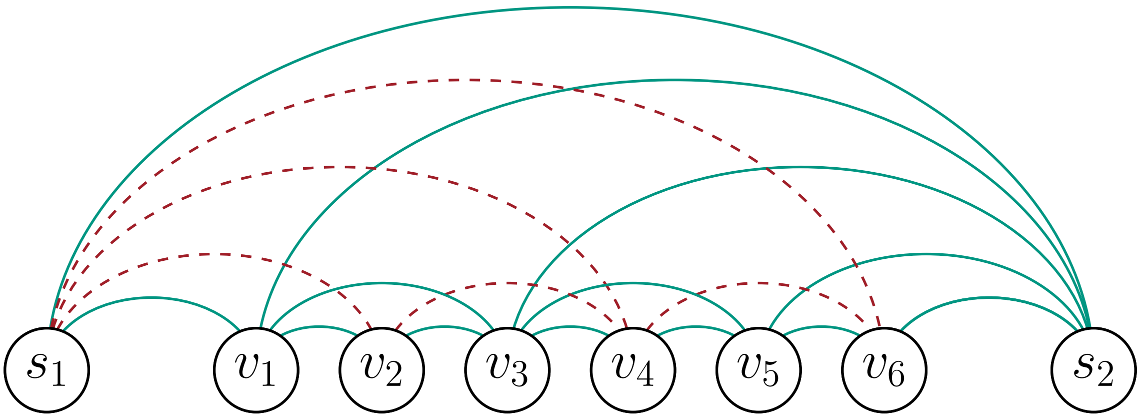

Figure 4.18: A 2-stack layout of the graph $P_{8,2}$ with all missing short edges added in.

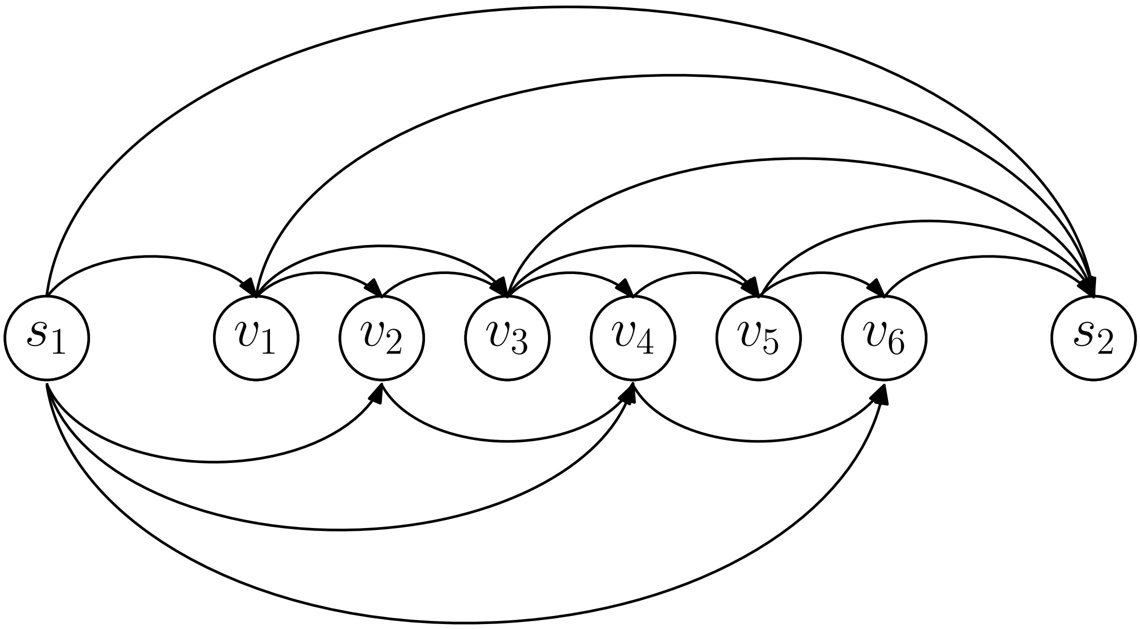

Figure 4.19: An upward planar drawing of the graph $P_{8,2}$ with all short edges and the edges directed according to the shown order of vertices σ.

5. Conclusions

For complete graphs and complete bipartite graphs $K_{m,n}$ with $m \ll n$ we were able to determine the forest stack number. Additionally, for outerplanar graphs, planar 3-trees and k-trees we were able to find tight upper bounds for the forest stack number. We did not determine the forest stack number for complete bipartite graphs in general since a tight upper bound for the stack number of complete bipartite graphs in general is not known. Hence, finding the forest stack number of complete bipartite graphs is still an open question. Currently the best known upper bound for the stack number of $K_{m,n}$ is $\lceil (m + 2n)/4 \rceil$[16]. Thus, finding a better upper bound for the forest stack number of complete bipartite graphs might help find or prove a smaller bound on the stack number of complete bipartite graphs.

Question 5.1. *Is there an upper bound for* $\mathrm{sn_f}(K_{m,n})$ *that is smaller than* $\min(m, n)$*?*

For the forest stack number of complete bipartite graphs $K_{m,n}$ we improved the trivial lower bound $\lceil (nm - n - m)/(n + m - 3) \rceil$ found by Bernhart and Kainen[3]. The lower bound for the forest stack number we found is $\lceil nm/(n + m - 1) \rceil$. In the best case, this bound is one large than the bound for the stack number. It is still an open question if this lower bound can be further improved.

Question 5.2. *Is there a lower bound for* $\mathrm{sn_f}(K_{m,n})$ *that is larger than* $\lceil nm/(n + m - 1) \rceil$*?*

Subhamiltonian graphs are the graphs with stack number at most 2[3]. Furthermore, any Hamiltonian cycle or order of vertices that can be made into a Hamiltonian cycle by adding edges, such that the graph remains planar, can be chosen as the order of vertices in a 2-stack layout. We have found a subhamiltonian graph with forest stack number 3. Additionally, Corollary 3.8 shows that at most four stacks are needed for a forest stack layout of subhamiltonian graphs. Using a fixed Hamiltonian cycle as the order of vertices, we found a subhamiltonian graph needing four stacks for a forest stack layout. However, if a subhamiltonian graph needing four stacks without a fixed order of vertices is found this would disprove Conjecture 1.1.

Question 5.3. *Is there a subhamiltonian graph G with* $\mathrm{sn_f}(G) = 4$*?*

We have found counterexamples to Conjecture 1.1 when using a fixed order of vertices or guaranteeing that a specific path in a graph is placed on a single stack. In general, the validity of Conjecture 1.1 is still an open question.

Question 5.4. *Is there a graph G with $\mathrm{sn}_{\mathrm{f}}(G) \geq \mathrm{sn}(G) + 2$?*

Getting closer to finding such a graph might be done by altering the counterexamples we found such that the restrictions we need to fulfill can be loosened.

Another possibility of attacking Question 5.4 that could be further researched is the introduction of a game for forest stack layouts. We consider a first player called Alice that iteratively introduces edges and vertices into a graph G such that the stack number of G stays at most k. The second player called Bob is given the newly added edges and vertices by Alice and has to insert them into a forest stack layout Γ of G that he is constructing. If for all additional edges and vertices that Alice introduces, Bob manages to keep the forest stack number of Γ below $k+2$, he wins. Alice can, however, react to the forest stack layout that Bob has constructed and wins if it is impossible for Bob to continue the layout on $k+1$ stacks. Thus, if a graph G with $\mathrm{sn}_{\mathrm{f}}(G) \geq \mathrm{sn}(G) + 2$ exists Alice can win the game by giving this graph to Bob, since it would be impossible for Bob to construct a forest stack layout of G using $\mathrm{sn}(G) + 1$ stacks.

In general, this game can always be won by Alice. She can give Bob a set of six vertices without any edges. Let a k-twist be k edges in a stack layout such that each of the k edges is crossing all other edges. For any order of vertices that Bob chooses for the forest stack layout, Alice can simply add three edges as shown in Figure 5.1 and construct a 3-twist. Thus, even though the graph has forest stack number 1, three stacks are needed for the forest stack layout Γ that Bob constructs.

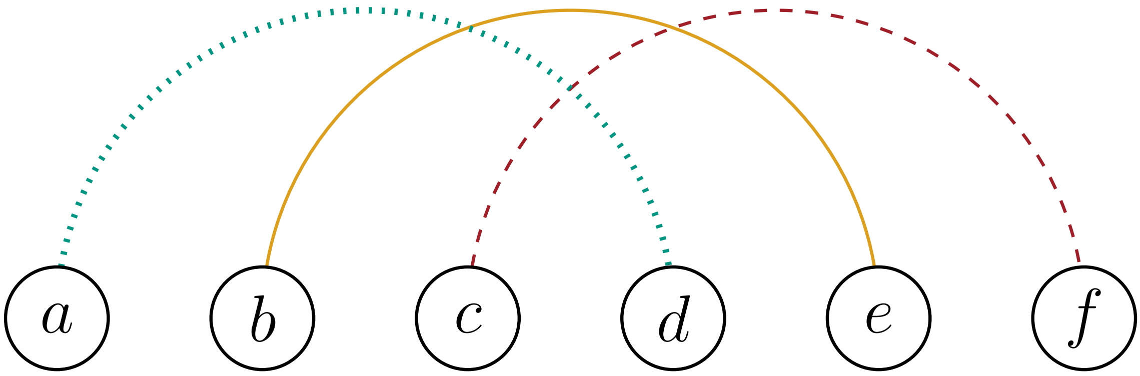

Figure 5.1: For any order of vertices the corresponding three edges can be added to create a 3-twist.

Thus, to make the game interesting, additional restrictions have to be placed on what edges Alice is allowed to add.

For example, Alice is only allowed to add edges such that any new edge is adjacent to a new vertex that is added simultaneously. Hence, the following question might bring us closer to determining the validity of Conjecture 1.1.

Question 5.5. *With rules as introduced above is there a way for Alice to always win the game?*

Considering graphs $F_{n,k}$ that need one additional forest stack, we introduce a different game. In this game Alice gives Bob a graph $F_{n,2}$ or, more generally, $F_{n,k}$. Bob then chooses an order of vertices σ. Next Alice is allowed to apply switching operations on the graph. We define the more generalized switching operation $\mathrm{sw}^{e}_{i,j}$ that removes the edges $s_j v_l$ with $i \leq l < i + e$ and inserts the edge $v_{i-1} v_{i+e}$ for a $j \in \{1, 2\}$ and an i with $2 \leq i \leq n - e - 2$. As an example the graph $\mathrm{sw}^{2}_{3,1}(F_{8,2})$ is shown in Figure 5.2. These new switching operations also do not increase the stack number. Alice is allowed to use the switching operations $\mathrm{sw}^1_{i,j} = \mathrm{sw}_{i,j}$ and $\mathrm{sw}^2_{i,j}$. After Alice has given Bob the switching operations, Bob has to find a forest 3-stack layout of the modified graph using the order of vertices σ that he chose before.

In this type of game, if Bob chooses the order of vertices $\sigma = s_1, v_1, v_2, \ldots, v_{n-2}, s_2$ then Theorem 4.5 tells us that no forest stack layout using three stacks exists if Alice constructs a $P_{n,k}$ with her switching operations. Thus, Alice wins. Else if Bob chooses the vertex ordering $\sigma' = v_2, v_4, \ldots, v_i, s_1, s_2, v_j, \ldots, v_3, v_1$ for which $P_{n,k}$ needs only three forest stacks, Alice can instead use the new switching operations. She can construct the graph shown in Figure 5.3. For this graph Figure 5.4 shows that four pairwise crossing edges exist when using the order of vertices σ'. Hence, Bob would also lose in this case. In the case that Bob groups together all odd path vertices and all even path vertices, a similar problem occurs. Hence, we pose the following open question.

Question 5.6. *For each order σ of vertices that Bob chooses, can Alice always find switching operations $\mathrm{sw}_{i,j}$ and $\mathrm{sw}^2_{i,j}$, such that Bob cannot construct a forest 3-stack layout using the order σ?*

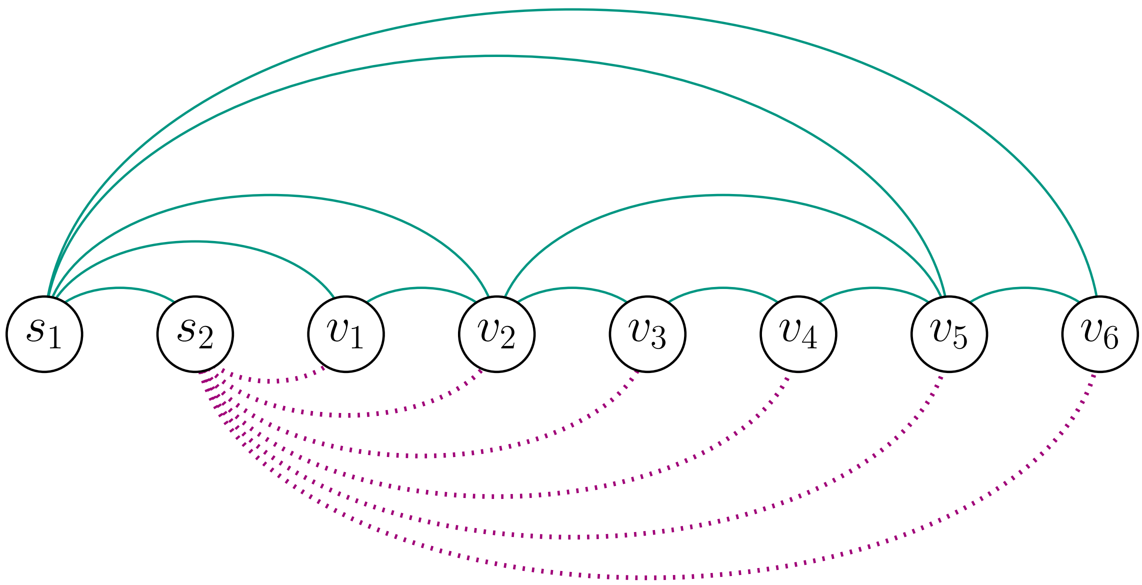

Figure 5.2: The graph $\mathrm{sw}^2_{3,1}(F_{8,2})$ with a 2-stack layout.

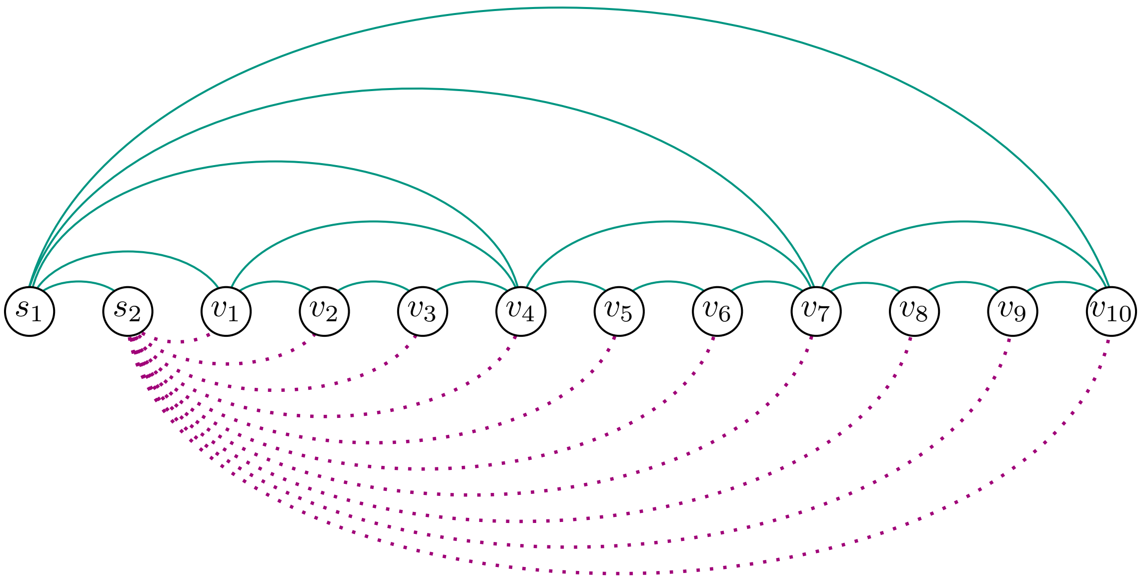

Figure 5.3: A graph that can be constructed by using switching operations $\mathrm{sw}_{i,j}^2$ on $F_{12,2}$.

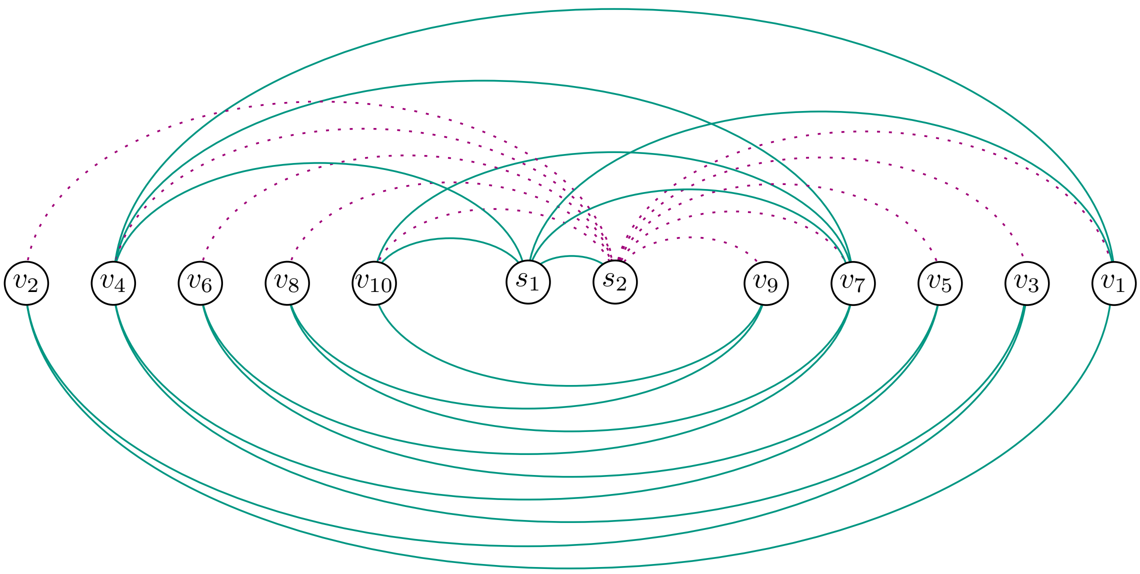

Figure 5.4: For this order of vertices the graph needs at least four stacks for a forest stack layout, since $s_1v_1, v_{10}v_7, v_8v_9$ and v_2s_2 are all pairwise crossing.

Bibliography

[1] J. M. Alam, M. A. Bekos, V. Dujmović, M. Gronemann, M. Kaufmann, and S. Pupyrev. On dispersable book embeddings. *Theoretical Computer Science*, 861:1–22, 2021. doi:0.1016/j.tcs.2021.01.035.

[2] M. A. Bekos, M. Kaufmann, F. Klute, S. Pupyrev, C. Raftopoulou, and T. Ueckerdt. Four pages are indeed necessary for planar graphs. *Journal of Computational Geometry*, 11(1):332–353, 2020. doi:10.20382/jocg.v11i1a12.

[3] F. Bernhart and P. 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.

[4] C. Binucci, G. D. Lozzo, E. D. Giacomo, W. Didimo, T. Mchedlidze, and M. Patrignani. Upward Book Embeddings of st-Graphs. In G. Barequet and Y. Wang, editors, *35th International Symposium on Computational Geometry (SoCG 2019)*, volume 129 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 13:1–13:22, Dagstuhl, Germany, 2019. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik. doi:10.4230/LIPIcs.SoCG.2019.13.

[5] H. Enomoto, T. Nakamigawa, and K. Ota. On the pagenumber of complete bipartite graphs. *Journal of Combinatorial Theory, Series B*, 71(1):111–120, 1997. doi:10.1006/jctb.1997.1773.

[6] Z. Galil, R. Kannan, and E. Szemerédi. On 3-pushdown graphs with large separators. *Combinatorica*, 9(1):9–19, 1989. doi:10.1007/BF02122679.

[7] M. R. Garey, D. S. Johnson, G. L. Miller, and C. H. Papadimitriou. The complexity of coloring circular arcs and chords. *SIAM Journal on Algebraic Discrete Methods*, 1(2):216–227, 1980. doi:10.1137/0601025.

[8] F. Gavril. The intersection graphs of subtrees in trees are exactly the chordal graphs. *Journal of Combinatorial Theory, Series B*, 16(1):47–56, 1974. doi:10.1016/0095-8956(74)90094-X.

[9] L. Heath. Embedding planar graphs in seven pages. In *25th Annual Symposium on Foundations of Computer Science, 1984.*, pages 74–83, 1984. doi:10.1109/SFCS.1984.715903.

[10] L. S. Heath and S. V. Pemmaraju. Stack and queue layouts of directed acyclic graphs: Part ii. *SIAM Journal on Computing*, 28(5):1588–1626, 1999. doi:10.1137/S0097539795291550.

[11] L. S. Heath, S. V. Pemmaraju, and A. N. Trenk. Stack and queue layouts of directed acyclic graphs: Part i. *SIAM Journal on Computing*, 28(4):1510–1539, 1999. doi:10.1137/S0097539795280287.

[12] P. Jungeblut, L. Merker, and T. Ueckerdt. A sublinear bound on the page number of upward planar graphs. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 963–978. doi:10.1137/1.9781611977073.42.

[13] J. L. Ganley and L. S. Heath. The pagenumber of k-trees is $\mathcal{O}(k)$. *Discrete Applied Mathematics*, 109(3):215–221, 2001. doi:10.1016/S0166-218X(00)00178-5.

[14] D. R. Lick and A. T. White. k-degenerate graphs. *Canadian Journal of Mathematics*, 22(5):1082–1096, 1970. doi:10.4153/CJM-1970-125-1.

[15] L. Merker and T. Ueckerdt. Local and union page numbers. In D. Archambault and C. D. Tóth, editors, *Graph Drawing and Network Visualization*, pages 447–459, Cham, 2019. Springer International Publishing. doi:10.1007/978-3-030-35802-0_34.

[16] D. J. Muder, M. L. Weaver, and D. B. West. Pagenumber of complete bipartite graphs. *Journal of Graph Theory*, 12(4):469–489, 1988. doi:10.1002/jgt.3190120403.

[17] M. Nöllenburg and S. Pupyrev. On families of planar dags with constant stack number. *arXiv preprint arXiv:2107.13658*, 2021. doi:10.48550/arXiv.2107.13658.

[18] R. Nowakowski and A. Parker. Ordered sets, pagenumbers and planarity. *Order*, 6(3):209–218, 1989. doi:10.1007/BF00563521.

[19] S. Overbay. *Generalized Book Embeddings*. PhD thesis, 05 1998. URL `https://www.researchgate.net/publication/267270871_Generalized_Book_Embeddings`.

[20] S. Pupyrev. Book embeddings of graph products, 2020. doi:10.48550/ARXIV.2007.15102.

[21] Z. Shao, Y. Liu, and Z. Li. Matching book embedding of the cartesian product of a complete graph and a cycle. 2020. doi:10.48550/ARXIV.2002.00309.

[22] J. Vandenbussche, D. B. West, and G. Yu. On the pagenumber of k-trees. *SIAM Journal on Discrete Mathematics*, 23(3):1455–1464, 2009. doi:10.1137/080714208.

[23] A. Wigderson. The complexity of the hamiltonian circuit problem for maximal planar graphs. Technical report, Tech. Rep. EECS 198, Princeton University, USA, 1982. URL `https://www.math.ias.edu/~avi/PUBLICATIONS/MYPAPERS/W82a/tech298.pdf`.

[24] M. 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.

[25] M. Yannakakis. Planar graphs that need four pages. *Journal of Combinatorial Theory, Series B*, 145:241–263, 2020. doi:10.1016/j.jctb.2020.05.008.