Noname manuscript No.
(will be inserted by the editor)

Scaffolding problems revisited: Complexity, Approximation and Fixed Parameter Tractable algorithms, and some special cases

Mathias Weller · Annie Chateau · Clément Dallard · Rodolphe Giroudeau

Received: 20 April 2016 / Accepted: 11 January 2018

Abstract This paper is devoted to new results about the scaffolding problem, an integral problem of genome inference in bioinformatics. The problem consists in finding a collection of disjoint cycles and paths covering a particular graph called the "scaffold graph". We examine the difficulty and the approximability of the scaffolding problem in special classes of graphs, either close to trees, or very dense. We propose negative and positive results, exploring the frontier between difficulty and tractability of computing and/or approximating a solution to the problem. Also, we explore a new direction through related problems consisting in finding a family of edges having a strong effect on solution weight.

Keywords complexity · approximation · lower bound · kernel · scaffolding

Mathias Weller
LIGM - CNRS UMR 8049, Marne-La-Vallée, France
E-mail: mathias.weller@u-pem.fr

Annie Chateau
LIRMM - CNRS UMR 5506, Montpellier, France
E-mail: annie.chateau@lirmm.fr

Clément Dallard
School of Computing, University of Portsmouth, UK
E-mail: clement.dallard@port.ac.uk

Rodolphe Giroudeau
LIRMM - CNRS UMR 5506, Montpellier, France
E-mail: rodolphe.giroudeau@lirmm.fr

1 Introduction

A lot of problems inspired by bioinformatics concerns may be formalized as combinatorial optimization problems on graphs. We focus in this paper on the genome scaffolding problem, which is of great importance when producing a genomic sequence from the real DNA molecule. Sequencing produces a huge amount of small sequences on the nucleotide alphabet $\{A, T, G, C\}$, called *reads*, whose overlaps are exploited to produce numerous sequences of various length, called *contigs*, during the *assembly* process. To complete the whole genome sequence, those contigs must be relatively ordered and oriented. In previous work on scaffolding, this problem has been modeled as a combinatorial problem on graphs which is, unfortunately, computationally hard [10]. Some methods use heuristic ways to simplify the graph [15], others use a decomposition of the problem into two separate steps (orienting and ordering), whose difficulty could be bypassed under certain restrictions [13]. A good presentation of the mainly used recent methods can be found in [18].

The following work is based on a simple formulation of input data and problem. We introduce the notion of *scaffold graph*, that is, an undirected graph for which an initial perfect matching is given. Vertices of the graph represent contig extremities. Edges in the matching represent the contigs. Since the aim of scaffolding is to decide an optimal orientation and order of these contigs, the scaffold graph count only vertices representing extremities of contigs, and that is why the matching is perfect. Other edges represent witnesses for the relative locations of the contigs. These latter edges are weighted by a flexible confidence measure that can be read from the sequencing data or mixed with, for example, ancestral support in a phylogenetic context. Then, the scaffolding problem consists in finding at most a number of σ_{p} paths and σ_{c} cycles that, together, cover all matching edges (contigs). We formally describe this problem in Section 2.

In previous works, we stated that the problem is $\mathcal{NP}$-complete, even in bipartite and planar graphs, and initiated the quest to the frontier between polynomial-time solvability and $\mathcal{NP}$-completeness [9, 10]. The beginnings of these results are presented in [24]. Aiming to circumvent the problem, we consider two classes of graphs, described in Section 2.

Exploring the structure of the scaffold graphs on real instances, we noted that many vertices of the scaffold graph have small degrees, leading to overall sparsity [23]. We aim to exploit this property to design algorithms tuned to instances occurring in practice. Since Scaffolding can be solved in polynomial time on graphs that are close to trees by measure of "treewidth" [23], we are interested in other distance measures to trees. To this end, we consider the class of graphs that can be turned into a (linear) forest by removing the edges of the given perfect matching M^* from it ("quasi forest"). In this paper, we consider Scaffolding on graphs G such that $G - M^*$ is a linear forest, a forest, a tree, or a path and show that the problem remains $\mathcal{NP}$-hard even for very restricted inputs. We reduce the $\mathcal{NP}$-complete Weighted 2SAT problem to it, allowing the inheritance of various hardness results of this problem. We

are also tackling the problem from the angle of the parameterized complexity, exploring the existence or non-existence of polynomial kernel for the problem in the hope of developing a fixed-parameter tractable algorithm. Section 3.5 describes how cross-composition leads to a negative result in quasi-forests.

We consider also dense graphs who we know are susceptible to polynomial-time approximation algorithms [9, 10]. We focus on dense graphs which are not entirely complete, yet allow encoding some structure, namely co-bipartite and split graphs. On co-bipartite graphs, the unweighted version of the scaffolding problem becomes polynomial-time solvable, which is a first step towards designing algorithms for the general problem on these graphs. We consider a slightly relaxed version of the problem to improve the known approximation algorithm on complete graphs [10] to a ratio of two.

To complete this overview of the various tracks allowing relevant results on the subject, we have also been interested in variations of the problem, inspired by the work on the minimum spanning tree and other classical combinatorial optimization problems [2, 3, 4]. These variants aim to detect critical subsets of edges or nodes in the graph, which can be used to detect a skeleton that we do not further question, and decrease the time consumption of an exact search on remaining edges. Unfortunately, we show that the problem to find such set of edges is also a difficult problem in Section 5.

The complexity and approximation results are respectively summarized in Table 1, Table 2 and Table 3.

The paper is organized as follows: Section 2 is devoted to a global presentation of problems, classes of graphs and technical issues. In Section 3 overview of the problems which remains hard, even with very strong constraints on parameters of the problem, structure of the graphs, or weights. After these depressing news, we focus on the lighter part of the work in Section 4, presenting polynomial-time special cases and approximation algorithms. Finally, in Section 5, we broaden our field of vision by considering several variants of the problem, showing initial (hardness) results.

2 Notation and Problem Description

Let $G = (V, E)$ be a graph. For a vertex set $V' \subseteq V$, let $G[V']$ denote the subgraph of G induced by V' and let $G - V' := G[V \setminus V']$. Slightly bending notation, we may consider an edge set $S \subseteq E$ as a graph which is implicitly defined as $(\bigcup_{e \in S} e, S)$. We further define $G - S := (V, E \setminus S)$. An edge set M^* of a graph is called *matching* if no two of its edges intersect, that is, $e_1 \cap e_2 = \varnothing$ for all distinct $e_1, e_2 \in M^*$. A matching M^* is *perfect* if it covers all the vertices, that is $V = \bigcup_{e \in M^*} e$. A pair (G, M^*) where M^* is a perfect matching on G is called a *scaffold graph*. For a matching M^* and a vertex u, we define $M^*(u)$ as the unique vertex v with $uv \in M^*$ if such a v exists, and $M^*(u) = \perp$, otherwise. We abbreviate $X - \{x\} =: X - x$ for any set X of elements of the same type as x. Slightly abusing notation, we identify a path with the set of its edges. A path p is *alternating* with respect to a matching M^* if, for all vertices u of p, also $M^*(u)$ is a vertex of p. Thus, alternating paths have an even

G	$\omega_{\max}=1$			$\omega_{\max}=0$	
	$\sigma_{\mathrm{p}},\sigma_{\mathrm{c}}>0$	$\sigma_{\mathrm{c}}=0$	$\sigma_{\mathrm{p}}=0$	$\sigma_{\mathrm{p}}>0$	$\sigma_{\mathrm{p}}=0$
bipartite bipartite, planar	$\mathcal{NP}$c (Thm. 1 & [10]) $\mathcal{NP}$c for SSCA (Thm. 3)				
co-bipartite	$\mathcal{NP}$c (Cor. 1), no $2^{o(m)}$-time alg. (Cor. 2), no $2^{o(\sqrt{n})}$-time alg. for SSCA (Cor. 3)			$\mathcal{P}$ (Thm. 9)	
split				$\mathcal{NP}$-h. (Thm. 2)	
quasi forest	$\mathcal{NP}$c, $\mathcal{W}[1]$-h wrt. k (Thm. 5) no $2^{o(m)}$- or $n^{o(k)}$-time algorithm (Cor. 5), for SSCA with $\ell_{\mathrm{p}}=1,\sigma_{\mathrm{c}}=1$ (Thm. 4)		$\mathcal{P}$ (Cor. 9)	open	$\mathcal{P}$ (Cor. 9)
linear quasi forest	No polynomial kernel w.r.t. treewidth (Thm. 7)		$\mathcal{P}$ (Cor. 9)	open	$\mathcal{P}$ (Cor. 9)

Table 1: Complexity results for Scaffolding on various graph classes depending on $\omega_{\max}$, σ_{p}, and σ_{c}.

G	max	min
clique, complete bipartite	2-approx. (Theorem 10)	$\notin \mathcal{APX}$ (Cor. 6 & [10]),
co-bipartite, split	open	$\omega_{\max}/\omega_{\min}$-approximation
quasi forest	no $n^{\frac{1}{2}-\epsilon}$-approx. (Cor. 7)	$\mathcal{APX}$-hard (Cor. 8)

Table 2: Complexity to approximate Scaffolding.

Problem	Complexity	Approximation
Strict Scaffolding(MV)		
even with $\sigma_{\mathrm{p}}=0$ and $\ell_{\mathrm{c}}=6$	$\mathcal{NP}$c (Th. 11)	
on bip. graphs, max degree four, $\ell_{\mathrm{c}}=12$	$\mathcal{NP}$c (Cor. 10)	min $\notin \mathcal{APX}$ (Cor. 12)
on planar bip. graphs, $\sigma_{\mathrm{p}}=1$, $\ell_{\mathrm{c}}=4$	$\mathcal{NP}$c (Cor. 11)	
Strict Scaffolding(EB)	$\mathcal{NP}$c (Cor. 13)	open
2-Scaffolding	$\mathcal{NP}$c (Th. 12)	open

Table 3: Complexity and approximation results for variant of Strict Scaffolding and Scaffolding problems (Section 5).

number of vertices. If M^* is clear from context, we do not mention it explicitly. For a function $\omega : E \to \mathbb{N}$ and a set $S \subseteq E$, we abbreviate $\sum_{e\in S}\omega(e) =: \omega(S)$ and we let $\omega_{\max} := \max_{e\in E}\omega(e)$. Thus, $\omega_{\max}=1$ (resp. $=0$) means that the weights can take only two values (resp. one value). The center of this work is the following problem.

Scaffolding (SCA)
Input: $G=(V,E)$, $\omega : E \to \mathbb{N}$, perfect matching M^* in G, $\sigma_{\mathrm{p}},\sigma_{\mathrm{c}},k \in \mathbb{N}$
Question: Is there an $S \subseteq E \setminus M^*$ such that $S \cup M^*$ is a collection of $\leq \sigma_{\mathrm{p}}$ alternating paths and $\leq \sigma_{\mathrm{c}}$ alternating cycles and $\omega(S) \geq k$?

If ω is uniform, that is, all edges have same weight, then we call the problem *unweighted* Scaffolding (USCA). The variant of the problem that asks for

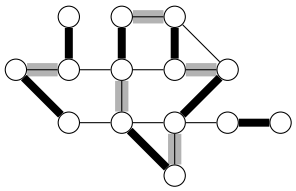


Fig. 1: An example of instance of Scaffolding. Matching edges are strong. With $(\sigma_{\mathrm{p}}, \sigma_{\mathrm{c}}) = (2, 2)$, it is positive for Scaffolding, but negative for Strict Scaffolding. A solution is given in gray.

exactly σ_{p} paths and *exactly* σ_{c} cycles is called Strict Scaffolding (SSCA). If we are looking for paths and cycles of *fixed lengths* ℓ_{p} and ℓ_{c}, we replace σ_{p} and σ_{c} by pairs $(\sigma_{\mathrm{p}}, \ell_{\mathrm{p}})$ and $(\sigma_{\mathrm{c}}, \ell_{\mathrm{c}})$ (length means the number of edges). We refer to the optimization variants of Scaffolding that ask to minimize or maximize $\omega(S)$ as Min Scaffolding and Max Scaffolding, respectively.

Classes of graphs. A graph is *bipartite* if it does not contain an odd cycle or, equivalently, if it admits a proper vertex two-coloring. It is usually given by a partition $X = X_1 \uplus X_2$. A *tripartite* graph is similarly defined as a graph which can be colored with three colors, so that no two endpoints of an edge have the same color. A graph is *co-bipartite* if its complement is bipartite. Thus, a co-bipartite graph can also be considered as a pair of disjoint cliques, with some edges between them. A *co-tripartite* graph is a graph whose complement is tripartite. For disjoint I and C, a graph $G = (I \cup C, E)$ such that I is an independent set, and C induces a clique in G, is called *split graph*. A scaffold graph (G, M^*) is called *quasi-forest* (resp. *quasi-tree* or *quasi-path*) if $G - M^*$ is a forest (resp. tree or path). The scaffold graph on Figure 1 is a quasi-forest.

Approximation algorithm. The main issue in approximation point of view consists in determining how close a polynomial-time algorithm can approach the optimal solution. Such polynomial-time algorithms producing solutions that are provably within a certain margin of the optimal are called *approximation algorithms*. Formally, the approximation-ratio of an algorithm A for a maximization problem is defined as $\rho := \max_I \frac{A(I)}{OPT(I)}$, where $OPT(I)$ is the optimal value of the instance I.

Lower bounds. The *Exponential-Time Hypothesis* [19, 20] states that there is some $c > 1$ such that n-variable 3−Satisfiability cannot be solved in $c^n \operatorname{poly}(n)$ time. Using polynomial reductions, it is possible to deduce some lower bounds on time-complexity for other problems.

Parameterized algorithms. An interesting way to tackle $\mathcal{NP}$-hard problems is parameterized complexity. A parameterized problem Q is a subset of $\Sigma^* \times \mathbb{N}$, where the second component is called the *parameter* of the instance. A *fixed-parameter tractable* ($\mathcal{FPT}$ for short) problem is a problem for which there exists an algorithm which, given $(x, k) \in \Sigma^* \times \mathbb{N}$, decides whether $(x, k) \in$

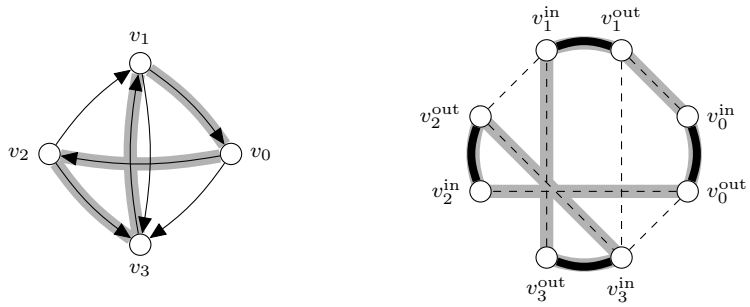


Fig. 2: Example of Construction 1, transforming the left instance of Directed Hamiltonian Cycle to the right graph with edges of M^* in bold and edges of the form $v_i^{\text{out}}v_j^{\text{in}}$ dashed. A corresponding solution is highlighted.

Q in time $f(k)|x|^{O(1)}$ for some computable function f. Such an algorithm becomes efficient with an hopefully small parameter. A *kernel* is a polynomial algorithm which, given $(x,k) \in \Sigma^* \times \mathbb{N}$, outputs an instance (x',k') such that $(x,k) \in Q \Leftrightarrow (x',k') \in Q$ and $|x'| + k' \leq f(k)$ for some computable function f. For decidable problems, the existence of a kernel is equivalent to the existence of an $\mathcal{FPT}$-algorithm. Nevertheless one can ask the function f to be a polynomial. If so, then the kernel is called a *polynomial kernel*. If a problem admits a polynomial kernel, then it roughly means that we can, in polynomial time, compress the initial instance into an instance of size $poly(k)$ which contains all the hardness of the instance.

3 Bad News: Hardness of Scaffolding

In this section, we focus on hard cases that we met during our attempts to determine the frontier between polynomiality and $\mathcal{NP}$-completeness. In all those attempts to simplify the problem, we use polynomial reduction from very well-known problems, such as Directed Hamiltonian Path, Partition into Triangles, or Weighted 2-SAT.

In following paragraphs, we use a reduction from the Directed Hamiltonian Path (resp. Directed Hamiltonian Cycle) [16] and some variations of this reduction. Thus, we define a basic construction that is a starting point to other constructions.

Directed Hamiltonian Path/ Cycle (DHP / DHC)
Input: A directed graph G without self loop
Question: Does G contain a simple path / cycle visiting all vertices?

Construction 1 *Let $G = (V = \{v_1, v_2, \ldots, v_n\}, A)$ be an instance of DHC. We construct $G' = (V_{in} \uplus V_{out}, E)$ as follows (see Figure 2).*

$$V_{in} := \{v_i^{in} \mid v_i \in V\} \qquad V_{out} := \{v_i^{out} \mid v_i \in V\}$$

$$E := \{v_i^{in} v_i^{out} \mid v_i \in V\} \cup \{v_i^{out} v_j^{in} \mid v_i v_j \in A\}.$$

Finally, let $M^* := \{v_i^{in} v_i^{out} \mid v_i \in V\}$*, let* $\omega : E \to \{0\}$*, and let* $k := 0$*.*

Lemma 1 *Let* G *be a digraph and let* G' *and* M^* *result from applying Construction 1 to* G*. Then,* G *has a directed Hamiltonian path (cycle) if and only if* M^* *can be covered by a single alternating path (cycle) in* G'*.*

Proof We show the lemma for the case of searching a *directed Hamiltonian cycle,* as the other case is analogous.

"⇒": Let $\mathcal{C}$ be a directed Hamiltonian cycle in G. Then, $S := \{v_i^{\text{out}} v_j^{\text{in}} \mid v_i v_j \in \mathcal{C}\}$ is a feasible solution, and $S \cup M^*$ is an alternating cycle covering M^*.

"⇐": Let S' be a matching in $G' - M^*$ such that $\mathcal{C}' := S' \cup M^*$ is an alternating cycle in G' covering M^*. Since $\mathcal{C}'$ contains $v_i^{\text{in}} v_i^{\text{out}}$ for each $v_i \in V$ (because they are all in M^*) and $\mathcal{C}'$ is a cycle, we know that S' contains n edges of the form $v_i^{\text{out}} v_j^{\text{in}}$ for $v_i v_j \in A$. Since S' is a matching in $G' - M^*$, no two of these edges are adjacent. Now, $\mathcal{C} := \{v_i v_j \mid v_i^{\text{out}} v_j^{\text{in}} \in S'\}$ is a collection of cycles covering all vertices of V in G. However, if $\mathcal{C}$ induces more than one cycle in G, then so does $S' \cup M^*$ in G'. Therefore, $\mathcal{C}$ is a Hamiltonian cycle in G. □

Since DHP and DHC remain $\mathcal{NP}$-complete for planar digraphs with maximum degree three [22], Lemma 1 implies that Scaffolding is $\mathcal{NP}$-complete on a very restricted class of graphs.

Theorem 1 *Unweighted* Scaffolding *with* $\sigma_{\text{p}} = 0$ *and* $\sigma_{\text{c}} = 1$ *is* $\mathcal{NP}$*-complete, even on bipartite planar graphs with maximum degree three.*

Finally, Construction 1 can be extended to split graphs by pairwise connecting all vertices v_i^{in} in G'. More precisely, we show that any number of edges can be added between any two vertices v_i^{in} and v_j^{in} of G'.

Lemma 2 *Let* $(G' = (V_{in} \uplus V_{out}, E), M^*, \omega, 0, 1)$ *be an instance produced by Construction 1, let* $X \subseteq \binom{V_{in}}{2}$ *and let* $G'' := (V_{in} \uplus V_{out}, E \uplus X)$*. Then,* $(G', M^*, \omega, 0, 1)$ *is a yes-instance of* Scaffolding *if and only if the instance* $(G'', M^*, \omega, 0, 1)$ *is.*

Proof Evidently, all alternating cycles in G' are also alternating in G'', so the "⇒" direction is trivial.

"⇐": Let C be an alternating cycle covering M^* in G''. Clearly, if C avoids X, it is also an alternating cycle in G' covering M^*. Thus, assume that $C \cap X$ contains an edge $v_i^{\text{in}} v_j^{\text{in}} =: e$. As C is an alternating cycle covering the perfect matching M^*, it touches each vertex of G' exactly once. Since $|V_{\text{in}}| = |V_{\text{out}}|$ and C contains v_i^{in} and v_j^{in} consecutively, there are two vertices v_r^{out} and v_s^{out} of V_{out} who occur consecutively (in the cyclic sense) in C, contradicting the fact that G'' does not contain edges between vertices of V_{out}. □

Theorem 2 *Let* $\mathfrak{G}$ *be a class of graphs such that each planar bipartite graph* $G = (U \uplus V, E)$ *has a supergraph in* $\mathfrak{G}$ *that results from adding edges between vertices of* U*. Unweighted* Scaffolding *with* $\sigma_{\text{p}} = 0$ *and* $\sigma_{\text{c}} = 1$ *is* $\mathcal{NP}$*-complete, even on* $\mathfrak{G}$*.*

Notably, Theorem 2 implies that unweighted Scaffolding is $\mathcal{NP}$-complete on the class of split graphs.

3.1 Hardness of Strict Scaffolding

In [10], we proved that, when the number of edges in the matching M^* is equal to $\sigma_p + 2\sigma_c$, then Strict Scaffolding can be solved in polynomial time, because all cycles have length four. We also proved that, for cycle length equal to six, it is $\mathcal{NP}$-complete. We investigate in this section the complexity of Strict Scaffolding in planar bipartite graphs for all even cycle-lengths (including four) and show that the problem remains $\mathcal{NP}$-complete.

Construction 2 *Given any $\sigma_p, \sigma_c, \ell_c \in \mathbb{N}$ with $\sigma_p > 0$ and ℓ_c even, augment the graph $G' = (V', E)$ and matching M^* resulting from applying Construction 1 to an instance of* Directed Hamiltonian Path *by adding $\sigma_p - 1$ disjoint alternating paths of length $|V'| - 1$ and σ_c disjoint alternating cycles of length ℓ_c.*

Theorem 3 Strict Scaffolding *is $\mathcal{NP}$-hard for any $\sigma_p, \sigma_c, \ell_c \in \mathbb{N}$ with $\sigma_p \neq 0$ and $\ell_c/2 \in \mathbb{N}$, even on planar bipartite graphs of maximum degree three.*

Proof Let G'' and M'' result from adding the disjoint alternating paths and cycles to G' and M^* and let G be the digraph from which Construction 1 computed G' and M^*. Clearly, if M^* can be covered by a single alternating path in G', then this path has length $|V'| - 1$ and we can cover M'' with exactly σ_p paths of length $|V'| - 1$ and σ_c cycles of length ℓ_c in G''.

On the other hand, if M'' can be thusly covered in G'' then, by construction, each connected component of G'' contains a single path or cycle of the solution. Thus, the component corresponding to G' is covered by a single alternating path or cycle. □

Note that the connected components in G'' can be joined by additional non-matching edges to make G'' connected, but the proof is much more technical.

3.2 Dense Graphs with Weights

Note that we can change Construction 1 such that $\omega : E \to \{1\}$ and $k := 2n$. This further enables us to add any number of edges of weight 0 without affecting the correctness argument, and the proof of Lemma 1 can be generalized to Scaffolding with $\omega_{\max} = 1$ on instances whose scaffold graph is a supergraph of a planar bipartite graph – provided all planar bipartite graphs are thusly represented. Since, by Theorem 1, Scaffolding is $\mathcal{NP}$-complete on the class of planar bipartite graphs, it remains $\mathcal{NP}$-complete on all classes $\mathfrak{G}$ of graphs such that all planar bipartite graphs have a supergraph in $\mathfrak{G}$ (such as split graphs and co-bipartite graphs).

Corollary 1 *Let $\mathfrak{G}$ be a class of graphs such that, for each planar bipartite graph G there is a supergraph of G in $\mathfrak{G}$. Then,* Scaffolding *is $\mathcal{NP}$-complete on $\mathfrak{G}$, even if $\sigma_{\mathrm{p}} = 0$, $\sigma_{\mathrm{c}} = 1$ and $\omega_{\max} = 1$.*

Construction 1 also implies subexponential lower bounds for our problems based on the widely believed complexity-theoretic hypothesis known as the "Exponential-Time Hypothesis[1]" (ETH, see [20, 25]). In fact, the lower bound is established for both Scaffolding and Strict Scaffolding from the fact that (planar) Directed Hamiltonian Cycle does not admit an $O(2^{o(m)})$-time algorithm [21, Theorem 3.5] and that Construction 1 only blows up the instance size *linearly*.

Corollary 2 *Let $\mathfrak{G}$ be a class of graphs such that, for each planar bipartite graph G there is a supergraph of G in $\mathfrak{G}$. Assuming ETH, there is no $O(2^{o(m)})$-time algorithm for* Scaffolding *on $\mathfrak{G}$, even if $\sigma_{\mathrm{p}} = 0$, $\sigma_{\mathrm{c}} = 1$ and $\omega_{\max} = 1$ (where m is the number of edges of the input graph).*

Corollary 3 *Let $\mathfrak{G}$ be a class of graphs such that, for each planar bipartite graph G there is a supergraph of G in $\mathfrak{G}$. Assuming ETH, there is no $O(2^{o(\sqrt{n})})$-time algorithm for* Strict Scaffolding*, even if σ_{p}, σ_{c}, and ℓ_{c} are arbitrary, fixed integers (with $\sigma_{\mathrm{p}} \neq 0$ and $\ell_{\mathrm{c}}/2 \in \mathbb{N}$).*

3.3 Sparse Graphs with Weights

The hardness of Scaffolding for dense graphs proved by Corollary 1 motivates the search for tractable cases among classes of sparse graphs. It is known that Scaffolding is polynomial-time solvable on graphs that are close to being a forest (constant treewidth) [23], so we consider a different sparsity measure here. We investigate whether Scaffolding becomes polynomial-time solvable if the result of removing the given perfect matching M^* from G forms a forest. We call this class of graphs "quasi forests". Remark that real scaffold graphs are not always quasi-forest, however this is a first step towards their structure. We start off by modifying Construction 1 to make the resulting graph a quasi tree (see Construction 3 and Figure 3). Unfortunately, this requires fixing the length of the sought Hamiltonian cycle. To circumvent this, we present another construction, reducing the $\mathcal{NP}$-complete Weighted 2SAT to Scaffolding, that does not require fixing the lengths.

Construction 3 *Let $(G' = (V', E), M^*, \omega, k)$ be the result of applying Construction 1 to an instance (V, A) of* Directed Hamiltonian Cycle*. We construct $G^\dagger = (V^\dagger, E^\dagger)$ and $M^\dagger$ from G' and M^* by replacing all edges of $E \setminus M^*$ by a path of three edges and adding the middle one to $M^\dagger$.*

Lemma 3 *The result of Construction 3 can be covered by $|A|-|V|$ alternating paths of length 1 and 1 alternating cycle of length $4|V|$ if and only if (V, A) has a directed Hamiltonian cycle.*

[1] The ETH states that there is a constant $c > 1$ such that n-variable 3SAT cannot be solved in $O(c^n)$ time.

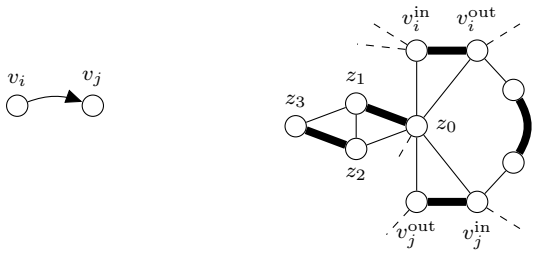


Fig. 3: How to make the result of Construction 3 connected.

Proof From each cover of M^* in G' with a single alternating cycle C, we can construct a cover of $M^\dagger$ in $G^\dagger$ by replacing each $v_i^{\text{out}}v_j^{\text{in}} \in C$ by the appropriate 3-path and adding all $|A| - |V|$ non-covered edges of $M^\dagger$ as alternating paths of length 1. For the other direction, assume that a cover of $M^\dagger$ in $G^\dagger$ uses a length-1 path to cover any of the edges of M^*. Then, strictly less than $|A|-|V|$ edges of $M^\dagger \setminus M^*$ are covered by such length-1 paths. By pigeonhole principle, some vertex in V' is incident with two non-matching edges that are used in the unique cycle $C^\dagger$ in the solution cover. Then, however, $C^\dagger$ is not alternating. □

Theorem 4 Strict Scaffolding *with* $\ell_{\text{p}} = 1$ *and* $\sigma_{\text{c}} = 1$ *is* $\mathcal{NP}$*-complete, even on quasi-forests.*

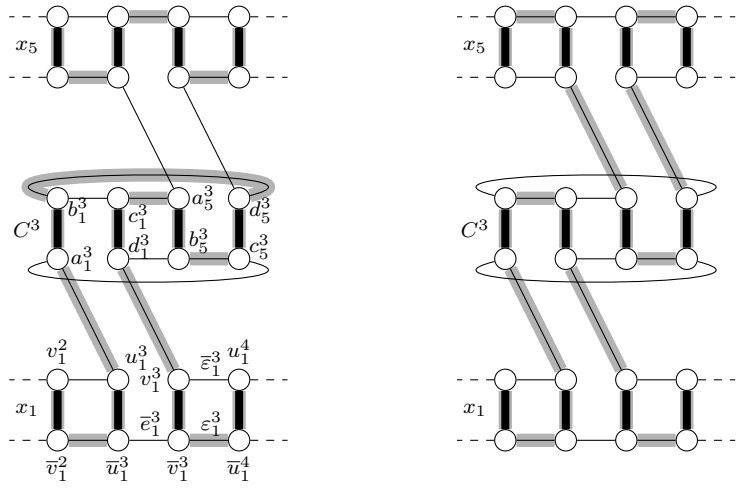
Theorem 4 can be extended to connected quasi-forests (thus "quasi-trees") by (a) adding new vertices z_0, z_1, z_2 and z_3, (b) connecting z_0 to all v_i^{in}, v_i^{out} in V' with $v_i \in V$, (c) adding the edges $\{z_1, z_2\}$, $\{z_1, z_3\}$, and $\{z_0, z_2\}$ and the matching edges $\{z_0, z_1\}$ and $\{z_2, z_3\}$. Clearly, the result is connected and if $\{z_0, z_1\}$ is contained in a cycle in the solution, then this cycle has length four (too short for meaningful instances of DHC (with $|V| > 2$)).

Also note that the proof of Lemma 3 requires fixed lengths of the paths and cycles and thus only applies to Strict Scaffolding. To show that Scaffolding is also hard on quasi-forests, we give another reduction, this time from the $\mathcal{NP}$-complete Weighted 2SAT problem [1].

Weighted 2SAT
Input: n variables x_i with weights $w_i \geq 0$, m size-two clauses, $k \in \mathbb{N}$
Question: Is there a truth assignment β s.t., $\sum_{i \mid \beta(x_i)=1} w_i \geq k$?

The optimization variants of Weighted 2SAT that ask to find a satisfying assignment β that minimizes or maximizes $\sum_{i \mid \beta(x_i)=1} w_i$ are called Min Weighted 2SAT and Max Weighted 2SAT, respectively.

Construction 4 *Let* (φ, k) *be an instance of* Weighted 2SAT *with* n *variables* $x_0, x_1, \ldots, x_{n-1}$ *and* m *clauses* $C^0, C^1, \ldots, C^{m-1}$*. We produce the following instance* $(G, \omega, M^*, n, 0, k)$ *of* Scaffolding *(see Figure 4), that we denote* $\Gamma(\varphi, k)$*. For each variable* x_i *and for each* $0 \leq j < m$*, introduce*



(a) $x_1 \vee \overline{x}_5$ satisfied by $x_1 = 1$.

(b) $x_1 \vee \overline{x}_5$ satisfied by $x_1 = 1$ and $x_5 = 0$.

Fig. 4: Example of Construction 4 for the clause $x_1 \vee \overline{x}_5$. Bold edges are in M^*. Gray paths are solution paths corresponding to the respective assignments.

– *vertices* u_i^j, $\overline{u}_i^j$, v_i^{j-1},$\overline{v}_i^{j-1}$,
– *edges* $u_i^j\overline{u}_i^j$,$v_i^{j-1}\overline{v}_i^{j-1}$ *that are also added to* M^*,
– *edges* $\overline{\varepsilon}_i^{j-1} := v_i^{j-1}u_i^j$, *and* $\varepsilon_i^{j-1} := \overline{v}_i^{j-1}\overline{u}_i^j$.
– *for* $j < m$, *if* C^j *contains* $\overline{x}_i$, *the edge* $e_i^j := u_i^j v_i^j$, *otherwise,* $\overline{e}_i^j := \overline{u}_i^j\overline{v}_i^j$.

For each clause C^j *on the variables* x_{ℓ_0} *and* x_{ℓ_1}, *introduce*

– *for each* $i \in \{\ell_0, \ell_1\}$,
  – *vertices* a_i^j, b_i^j, c_i^j, d_i^j
  – *edges* $a_i^j b_i^j$ *and* $c_i^j d_i^j$ *that are added to* M^* *and* $b_i^j c_i^j$,
  – *if* C^j *contains* $\overline{x}_i$, *edges* $\overline{u}_i^j a_i^j$, $\overline{v}_i^j d_i^j$, *otherwise,* $u_i^j a_i^j$, $v_i^j d_i^j$,
– *edges* $a_{\ell_0}^j c_{\ell_1}^j$, $c_{\ell_0}^j a_{\ell_1}^j$, $b_{\ell_0}^j d_{\ell_1}^j$, *and* $d_{\ell_0}^j b_{\ell_0}^j$.

Finally, set $\omega(\varepsilon_i^{m-1}) := 1$ *for each variable* x_i *and set the weights of all other edges to* 0.

Lemma 4 *Construction 4 is correct, that is,* φ *has a satisfying assignment of weight* k *if and only if* $(G, \omega, M^*, n, 0, k)$ *is a yes-instance of* Scaffolding.

Proof "⇒": Let β denote a solution for (φ, k). Then, we construct a solution S for $(G, \omega, M^*, n, 0, k)$ as follows. For each variable x_i and each $0 \leq j \leq m$, if $\beta(x_i) = 1$ then include $\{e_i^j, \varepsilon_i^j\} \cap E(G)$ in S, otherwise include $\{\overline{e}_i^j, \overline{\varepsilon}_i^j\} \cap E(G)$ in S. For all clauses C^j, if exactly one of its literals is true, include edges according to Figure 4a, if both its literals are true, include edges according to Figure 4b in S. Then, $S \cup M^*$ contains exactly 1 alternating path for each of the n variables and, since $\varepsilon_i^{m-1} \in S$ for each x_i with $\beta(x_i) = 1$, the weight of S equals the weight of β, which is at least k.

"⇐": Let S be a solution for $(G, \omega, M^*, n, 0, k)$. Note that $S \cup M^*$ contains at most n paths and no cycles. Since $S \cup M^*$ does not contain cycles, for each $i < n$ and $j \leq m$ we have $\varepsilon_i^j \notin S$ or $\overline{\varepsilon}_i^j \notin S$. This implies that, for

each $i < n$ there is a path ending at u_i^m or $\overline{u}_i^m$ and there is a path ending at v_i^{-1} or $\overline{v}_i^{-1}$. Since there are at most n paths in $S \cup M^*$, the "or" above are exclusive and all other vertices have degree exactly two in $S \cup M^*$, implying that

$$\text{all other vertices are incident to exactly one edge in } S. \tag{1}$$

Next, we show for all $i < n$ and $j < m$ that

$$u_i^j a_i^j \in S \iff v_i^j d_i^j \in S. \tag{2}$$

To show $u_i^j a_i^j \in S \Rightarrow v_i^j d_i^j \in S$, assume $u_i^j a_i^j \in S$ and $v_i^j d_i^j \notin S$. Then, either $b_i^j c_i^j \in S$ or $b_i^j d_\ell^j \in S$ for some $\ell \neq i$. In the first case, we have $d_i^j b_\ell^j \in S$ and, thus, c_ℓ^j cannot have an incident edge in S without violating (1). In the second case, note that the only edges incident to c_i^j and d_i^j that could be in S without violating (1) are $c_i^j a_\ell^j$ and $d_i^j b_\ell^j$, respectively. However, if both are in S, then $S \cup M^*$ contains a forbidden cycle. The direction $v_i^j d_i^j \in S \Rightarrow u_i^j a_i^j \in S$ can be shown analogously.

Next, we show for each $i < n$ and $j \leq m$, that $\varepsilon_i^j \in S$ or $\overline{\varepsilon}_i^j \in S$, implying

$$\overline{\varepsilon}_i^j \in S \iff \varepsilon_i^j \notin S. \tag{3}$$

This is easy to see for $j = m$ since one of u_i^m and $\overline{u}_i^m$ has degree 2 in $S \cup M^*$. So let the claim hold for $j+1$ but not for j, that is, $\varepsilon_i^j, \overline{\varepsilon}_i^j \notin S$. If x_i is not contained in C^j, this means that both e_i^{j+1} and $\overline{e}_i^{j+1}$ are in S, forming a forbidden cycle. Thus, by symmetry, let C^j contain x_i non-negated. Then, S contains both $\overline{e}_i^{j+1}$ and $u_i^{j+1} a_i^{j+1}$ and, by (2), also $v_i^{j+1} d_i^{j+1}$. Then, by (1), none of ε_i^{j+1} and $\overline{\varepsilon}_i^{j+1}$ are in S, contradicting that the claim holds for $j+1$. Thus, (3) holds by induction.

Next, we show for each $i < n$ and $j < m$ that

$$\varepsilon_i^j \in S \iff \varepsilon_i^{j-1} \in S. \tag{4}$$

Note that, by (3) it is sufficient to prove $\varepsilon_i^j \in S \Rightarrow \varepsilon_i^{j-1} \in S$ and $\overline{\varepsilon}_i^j \in S \Rightarrow \overline{\varepsilon}_i^{j-1} \in S$. Consider some $i < n$ and $j < m$ such that $\varepsilon_i^j \in S$. Then, by (1), we have $\overline{v}_i^j d_i^j \notin S$ and $\overline{e}_i^j \notin S$. By (2), it follows that $\overline{u}_i^j a_i^j \notin S$ and, thus, by (1), $\varepsilon_i^{j-1} \in S$. Note that $\overline{\varepsilon}_i^j \in S \Rightarrow \overline{\varepsilon}_i^{j-1} \in S$ can be shown analogously.

Finally, we define the assignment β for φ as $\beta(x_i) = 1 \iff \varepsilon_i^{m-1} \in S$. Then, since $\omega(\varepsilon_i^{m-1}) = 1$ for all $i < n$, we know that β assigns 1 to at most k variables. It remains to show that β satisfies φ. To this end, assume that a clause C^j is not satisfied and let x_i and x_ℓ denote the variables occurring in C^j. Note that at least one of the edges $u_i^j a_i^j$, $\overline{u}_i^j$, $u_\ell^j a_\ell^j$, and $\overline{u}_\ell^j a_\ell^j$ is in S since, otherwise, none of the n paths ending in the variable gadgets can visit the clause gadget of C^j. Since the prove is symmetric in all four cases, let us assume $u_i^j a_i^j \in S$. Then, C^j contains x_i non-negated. By (1), we have $\overline{\varepsilon}_i^{j-1} \notin S$, which, by (3) implies $\varepsilon_i^{j-1} \in S$ and, by (4), we arrive at $\varepsilon_i^{m-1} \in S$. Thus, $\beta(x_i) = 1$ and, thus, C^j is satisfied by β. □

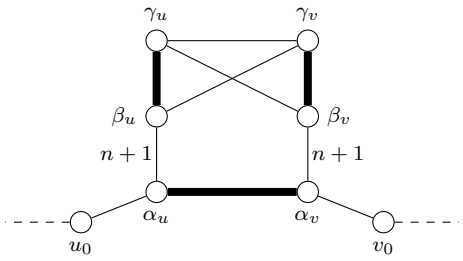


Fig. 5: How to merge any two paths in the result of Construction 4.

Since Weighted 2SAT is known to be $\mathcal{W}[1]-$hard with respect to k (that is, an algorithm that is exponential only in k is unlikely to exist [14]), by Lemma 4, so is Scaffolding.

Theorem 5 Scaffolding *with* $\omega_{\max} = 1$ *and* $\sigma_c = 0$ *is* $\mathcal{NP}$*-hard and* $\mathcal{W}[1]$*-hard w.r.t.* k*, even on bipartite graphs* G *with* $G - M^*$ *being a linear forest.*

Construction 4 can be modified such that $G - M^*$ is connected. We show how to merge any two paths $p = (v_0, v_1, \ldots)$ and $q := (u_0, u_1, \ldots)$ in $G - M^*$ (see Figure 5): (a) add new vertices α_j, β_j, γ_j for all $j \in \{u, v\}$, (b) add matching edges $\alpha_u\alpha_v$, $\beta_u\gamma_u$, $\beta_v\gamma_v$, (c) add non-matching edges $\beta_u\gamma_v$, $\beta_v\gamma_u$, $\gamma_u\gamma_v$, $v_0\alpha_v$, $u_0\alpha_u$ of weight 0, and (d) add non-matching edges $\alpha_u\beta_u$, $\alpha_v\beta_v$ of weight $n+1$. Finally, we ask for a solution of weight $2(n+1)+k$ containing $\sigma_p := n+1$ paths. Then, since all solutions have to contain the two heavy edges $\alpha_u\beta_u$ and $\alpha_v\beta_v$, no solution can contain either $u_0\alpha_u$ or $v_0\alpha_v$ and, thus, any solution contains a solution for the original instance. Repeating this operation lets us merge all paths in $G - M^*$.

Corollary 4 Scaffolding *with* ω *threestate and* $\sigma_c = 0$ *is* $\mathcal{NP}$*-hard and* $\mathcal{W}[1]$*-hard with respect to* k*, even on bipartite graphs* G *with* $G - M^*$ *being a path.*

In analogy with Corollary 2, Construction 4 implies subexponential-time lower bounds for exact algorithms. To this end, we modify Construction 4 slightly such that the gadget for each variable x_i contains a "module" (subgraph induced by u_i^j, $\overline{u}_i^j$, v_i^j, and $\overline{v}_i^j$) *only for the clauses it is actually contained in*. Thus, the number of vertices and edges in the produced instance can be bounded linearly in the number of clauses of the Weighted 2SAT instance. Then, since Independent Set (which is a special case of Weighted 2SAT) does not have a $2^{o(m)}$-time algorithm [20] (with m denoting the number of edges in the input graph), Scaffolding does not have a $2^{o(m)}$-time algorithm (unless the ETH fails). By the same argument, Independent Set not having an $n^{o(k)}$-time algorithm [11] implies that Scaffolding does not have an $n^{o(k)}$-time algorithm (unless $\mathcal{W}[1] = \mathcal{FPT}$).

Corollary 5
1. *Assuming ETH,* Scaffolding *cannot be solved in* $2^{o(m)}$ *time, and,*
2. *assuming* $\mathcal{W}[1] \neq \mathcal{FPT}$, Scaffolding *cannot be solved in* $n^{o(k)}$ *time,*

even if $\sigma_c = 0$, $\omega_{\max} = 1$, *and* $G - M^*$ *is a linear forest.*

Note that all results in this section hold for any numbers $\sigma_p \geq n$ and $\sigma_c \geq 0$ since we can add more paths artificially by adding isolated matching edges and we can add more cycles by adding new 4-cycles with heavy non-edges. Clearly, the isolated matching edges must constitute isolated paths. Further, if any isolated 4-cycle is covered by a path, it cannot collect all the weights of the heavy edges of the 4-cycle, which is necessary for a solution.

3.4 Approximation Hardness

We derive inapproximability of Scaffolding from Construction 1 (see Page 6).

Corollary 6 *Let* $\mathfrak{G}$ *be a class of graphs such that, for each planar bipartite graph* G *there is a supergraph of* G *in* $\mathfrak{G}$*. For all* $\rho \in \mathbb{Q}$ *with* $\rho > 1$, Min Scaffolding *on* $\mathfrak{G}$ *is* $\mathcal{NP}$*-hard to approximate to within a factor of* ρ*, even if* $\sigma_p = 0$ *and* $\sigma_c = 1$ *and* $\omega_{\max} = 1$.

Proof Suppose that there is a polynomial-time approximation algorithm $\mathfrak{A}$ for this problem with approximation ratio $\rho > 1$. Let $G = (V, E)$ be an instance of Directed Hamiltonian Cycle with $|V| = n$. We use Construction 1 to construct a bipartite graph G' with matching M^* and set of edges E'. Then, we let $\omega : E' \to \mathbb{N}$ such that $\omega(E' \setminus M^*) = 0$ and set $k := 0$. Then, we can add any number of edges of weight 1 and no solution computed by $\mathfrak{A}$ can contain any of these edges. Then, replacing $4n$ by 0 in Subsection 3.2 yields a proof for Corollary 6. Indeed, if G has a Hamiltonian cycle, then $\mathfrak{A}$ finds a solution of weight $\rho \cdot 0 = 0$. Conversely, if G does not have a Hamiltonian cycle, then at least one edge of weight 1 must be taken in a solution produced by $\mathfrak{A}$. Thus, $\mathfrak{A}$ decides the $\mathcal{NP}$-complete Directed Hamiltonian Cycle problem in polynomial time. □

While there is little hope of finding a constant-factor polynomial-time approximation algorithm for Min Scaffolding, there is a linear-time algorithm with approximation ratio $\frac{\omega_{\max}}{w_{\min}}$ (where $\omega_{\max}$ and $\omega_{\min}$ denote the respective maximum and minimum edge weights) on complete bipartite graphs with $\sigma_p = 0$ and $\sigma_c = 1$. This algorithm repeatedly chooses the lowest weighted edge that does not close the cycle.

Since Max Weighted 2SAT is $\mathcal{NP}$-hard to approximate to within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$ [1, 17] and the number of vertices in the instance produced by Construction 4 is bounded by the square of the number of variables, we conclude that, in contrast to the the factor-2 approximation for Max Scaffolding in complete (bipartite) graphs presented in Section 4.1, the problem is hard to approximate in bipartite quasi-forests.

Corollary 7 Max Scaffolding *with* $\omega_{\max} = 1$ *and* $\sigma_c = 0$ *is* $\mathcal{NP}$*-hard to approximate to within a factor of* $n^{\frac{1}{2}-\epsilon}$ *for any* $\epsilon > 0$*, even on bipartite graphs* G *with* $G - M^*$ *being a linear forest.*

For the minimization version, Min Scaffolding, we derive approximation hardness as well. To see this, note that Construction 4 is an S-reduction (see [12]) and Min Weighted 2SAT is $\mathcal{APX}$-complete [1]. Thus, Min Scaffolding is $\mathcal{APX}$-hard.

Corollary 8 Min Scaffolding *is $\mathcal{APX}$-hard even on bipartite cubic graphs G with $G - M^*$ being a linear forest, $\omega_{\max} = 1$ and $\sigma_c = 0$.*

Curiously, the approximation hardness result for Min Scaffolding is weaker than that for Max Scaffolding, which contrasts earlier observations on general graphs [10]. Thus, we suspect that Corollary 8 can be strengthened to at least the same hardness-level as we have for Max Scaffolding (Corollary 7).

3.5 Kernelization Hardness

In this section, we show that Scaffolding does not admit a polynomial-time preprocessing ("kernelization") that shrinks instances to polynomial-size measured in the treewidth of the input scaffold graph. The treewidth of a graph G is a measure of how treelike G is and it is defined as one less than the size of a largest bag in a "tree decomposition" minimizing this size. A *tree decomposition* of $G = (V, E)$ is a tree T whose nodes are subsets of V ("bags") such that each edge of E is contained in one of the bags and, for each $v \in V$, the subgraph of T induced by the bags containing v is connected (see [5, 6] for details). The treewidth is a popular graph measure that often allows designing fast algorithms (also for Scaffolding [23]) but no polynomial-size kernels.

In order to rule out polynomial kernels, we will use the recent technique of cross-composition [7]. Roughly speaking, a cross-composition is a polynomial reduction from t instances of a (non-parameterized) problem A to a single instance of a parameterized problem B such that the constructed instance is positive if and only if one of the input instances is positive. In addition, the parameter of the constructed instance must be bounded polynomially in the maximum size of the input instances and a logarithm of t. It is known that if A is $\mathcal{NP}$-hard and A cross-composes into B, then B cannot admit a polynomial kernel unless $\mathcal{NP} \subseteq co\mathcal{NP}/\operatorname{poly}$, where $co\mathcal{NP}/\operatorname{poly}$ is the class of decision problems refutable by a family of polynomial-size Boolean circuits.

The cross-composition framework allows us to assume some properties of the t input instances in form of an equivalence relation. We will use this equivalence relation to force all of the t input instances to have the same bound σ_p of requested paths and the same sought solution weight k.

Definition 1 (Polynomial equivalence relation [7]) An equivalence relation $\mathcal{R}$ on Σ^* is called a polynomial equivalence relation if both following conditions hold:

- There is an algorithm that given two strings $x, y \in \Sigma^*$, decides whether x and y belong to the same equivalence class in $(|x| + |y|)^{O(1)}$ time.
- For any finite set $S \subseteq \Sigma^*$, the equivalence relation $\mathcal{R}$ partitions the elements of S into at most $(max_{x \in S}|x|)^{O(1)}$ classes.

Definition 2 (OR-cross-composition (resp. AND) [7]) Let $L \subseteq \Sigma^*$ be a set and let $Q \subseteq \Sigma^* \times \mathbb{N}$ be a parameterized problem. We say that L OR-cross-composes (resp. AND-cross-composes) into Q if there is a polynomial equivalence relation $\mathcal{R}$ and an algorithm which, given t strings belonging to the same equivalence class of $\mathcal{R}$, computes an instance $(x^*, k^*) \in \Sigma^* \times \mathbb{N}$ in time polynomial in $\sum_{i=1}^{t} |x_i|$ such that:

- $(x^*, k^*) \in Q \Leftrightarrow x_i \in L$ for some $1 \leq i \leq t$ (resp. for all $1 \leq i \leq t$)
- k^* is bounded by a polynomial in $max_{i=1}^{t}|x_i| + \log t$

Theorem 6 ([7]) *If a set $L \subseteq \Sigma^*$ is $\mathcal{NP}$-hard and L AND-cross-composes into the parameterized problem Q, then there is no polynomial kernel for Q unless $\mathcal{NP} \subseteq co\mathcal{NP}/\operatorname{poly}$.*

Equipped with these tools, we introduce the problem that we will cross compose into Scaffolding parameterized by the treewidth of the input scaffold graph. The problem is a variant of Weighted 2SAT, called *Global Verification*, in which we want to find a solution for a given value k, knowing that no solution exists for $k-1$.

Weighted 2SAT(GV)
Input: a 2-CNF formula φ on variables X, weights $w : X \to \mathbb{N}$, $k \in \mathbb{N}$ such that φ cannot be satisfied with an assignment of weight $k-1$
Question: Can φ be satisfied by a weight-k assignment?

Lemma 5 *The problem* Weighted 2SAT(GV) *is $\mathcal{NP}$-hard.*

We proof that the special case of Weighted 2SAT(GV) in which all variables occur nonnegated is $\mathcal{NP}$-hard. This problem is equivalent to the following one.

Vertex Cover(GV)
Input: graph $G = (V, E)$, $k \in \mathbb{N}$ s.t. G has no vertex cover of size $k-1$
Question: Does G have a size-k vertex cover?

Proof (of Lemma 5) We use the classical reduction of 3SAT to Vertex Cover (see [16]): Let ψ be a instance of 3SAT with n variables and m clauses. For each variable x_i, create two vertices v_i and $\overline{v_i}$ that are connected by an edge and for each clause C_j, create a triangle (a clique of three vertices). Then, for each variable x_i of C_j, connect one of the vertices of the triangle to v_i if x_i is a literal of C_j and to $\overline{v_i}$ if $\neg x_i$ is a literal of C_j. This connection can easily be made such that all vertices of the clause triangle have degree three. Let the resulting graph be called G and let $k := n + 2m$.

Clearly, the vertices of G decompose into m triangles and n edges and, thus, G does not have a vertex cover of size $k-1$. Further, a vertex cover of size k can be attained if and only if each triangle contains a vertex that is not in the vertex cover. Let u be such a vertex in the gadget of C_j. Since u is adjacent to a vertex in a variable gadget, we know that this vertex is in the vertex cover and its literal can be used to satisfy C_j. Likewise, one can see that choosing vertices of the variable gadgets into a vertex cover according to a satisfying assignment allows covering all triangles with $2m$ vertices. □

Lemma 6 *For each $1 \leq i < t$, let (φ_i, k) be an instance of* Weighted 2Sat(GV) *and let $(G_i, \omega_i, M_i^*, \sigma_\mathrm{p}, 0, k)$ be the result of applying Construction 4 to (φ_i, k). Let G and M^* be the disjoint unions of all G_i and all M_i^*, respectively. Then, $(G, \bigcup_i \omega_i, M^*, t \cdot \sigma_\mathrm{p}, 0, t \cdot k)$ is a yes-instance of* Scaffolding *if and only if each (φ_i, k) is a yes-instance of* Weighted 2Sat(GV).

Proof Recall that no φ_i can be satisfied with an assignment of weight at most $k-1$. Moreover, for each i, Lemma 4 implies that (φ_i, k) is a yes-instance of Weighted 2Sat(GV) if and only if M_i^* can be covered by at most σ_p alternating paths with weight $\geq k$ in G_i.

"$\Leftarrow$": Let $(\varphi_i, k) \in$ Weighted 2Sat(GV) for each i. By Lemma 4, each M_i^* can be covered by at most σ_p alternating paths with weight $\geq k$ in G_i. Thus, the union of these paths covers M^* with at most $t \times \sigma_\mathrm{p}$ alternating paths with weight $\geq t \times k$ in G.

"$\Rightarrow$": Let S be a collection of $\leq t \times \sigma_\mathrm{p}$ alternating paths of weight $\geq t \times k$ covering M^* in G. Since none of the φ_i can be satisfied with an assignment of weight $k-1$, no M_i^* can be covered with $\leq \sigma_\mathrm{p}$ alternating paths in G_i. Thus, the restriction of S to any G_i has weight exactly k. Observe that, by Construction 4, each M_i^* needs at least σ_p paths to be covered in G_i (see the first lines of the "$\Leftarrow$"-direction of the proof of Lemma 4) and no path of S can contain vertices of G_i and G_j for any $i \neq j$. Thus, for each i, the restriction of S to G_i covers M_i^* with σ_p paths of weight k and, by Lemma 4, each φ_i has a satisfying assignment of weight k. □

Now, it is evident that the treewidth of the disjoint union of two graphs of treewidth at most w is at most w and the treewidth of each instance G_i is at most the size of G_i whose size can clearly be bounded in $|\varphi_i|$. Thus, Theorem 6 implies the following.

Theorem 7 Scaffolding *in linear quasi-forests does not admit a polynomial kernel parameterized by treewidth, unless $\mathcal{NP} \subseteq co\mathcal{NP}/\operatorname{poly}$.*

Efforts to show an analogue to Theorem 7 for the parameter σ_p failed (see also Corollary 9). An indication for this is the unlikeliness of being able to join alternating paths with uncertain endpoints.

Definition 3 A connector $\oplus$ is a polynomial-time operator such that for each quasi-forest (G, M^*),

1. $\oplus(G, M^*)$ is a quasi-forest and
2. for each $\sigma_\mathrm{p} \in \mathbb{N}$, it holds that M^* can be covered by σ_p alternating paths in G if and only if $\oplus(G, M^*)$ can be covered by $\sigma_\mathrm{p} - 1$ alternating paths.

Per intuition, arbitrarily reducing σ_p in quasi-forests in polynomial time combined with Corollary 9 contradicts the $\mathcal{NP}$-hardness of Scaffolding.

Theorem 8 *There is no connector $\oplus$ unless $\mathcal{P} = \mathcal{NP}$.*

4 Good News: What can be Done in Polynomial Time

4.1 Solving Dense Graphs

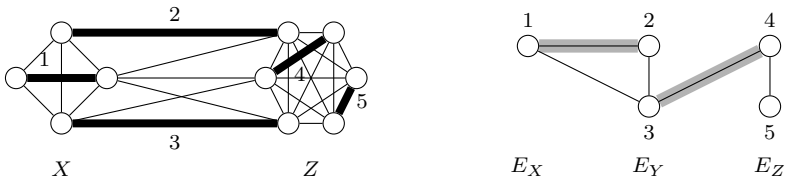
The complexity of Scaffolding may depend on the number of weights we allow in the input, to the point where $\omega_{\max} = 0$, which we call the *unweighted* case. Noticeably, unweighted Strict Scaffolding on cliques is trivial: it suffices to compare $|M^*|$ to $\sigma_{\mathrm{p}} + 2\sigma_{\mathrm{c}}$ (see [10]). A natural question is whether it is possible to extend this result to dense graph classes containing cliques, like co-bipartite. We present here a first result on co-bipartite graphs contrasting $\mathcal{NP}$-completeness on bipartite graphs (see Corollary 1).

Theorem 9 *Unweighted* Strict Scaffolding *can be solved in time complexity $O(n+m)$ on co-bipartite graphs.*

In the following, let G be a co-bipartite graph and let M^* be a perfect matching in G. Let H be the graph on the vertex set M^* that contains an edge $\{uv, xy\}$ if and only if $G[uvxy]$ contains an alternating cycle of length four. Note that, since G is co-bipartite with partition $X \uplus Z = V(G)$, we know that H is co-tripartite on partition $E_X \uplus E_Y \uplus E_Z = V(H)$ with $E_X = \binom{X}{2} \cap M^*$, $E_Z = \binom{Z}{2} \cap M^*$, and $E_Y = \{uv \mid u \in X \wedge v \in Z\} \cap M^*$. In the following, we call a matching M_H *nice* if, whenever M_H does not cover any $e \in E_X$ (or $e \in E_Z$), then there is no $\{e_1, e_2\} \in M_H$ with $e_1 \in E_X$ (or $e_1 \in E_Z$) and $e_2 \notin E_X$ (or $e_2 \notin E_Z$). Note that any matching M_H can be made nice in $O(|M_H|)$ time by flipping at most 2 alternating paths.

In the following, let M_H be the result of 1. joining any maximal matchings in $H[E_X]$, $H[E_Y]$, and $H[E_Z]$, 2. flipping any one augmenting path, and 3. making the resulting matching nice. Note that such an M_H can be found in linear time. Furthermore, since each of E_X, E_Y and E_Z induces a clique in H, there are at most 3 uncovered vertices after Step 1 and at most one uncovered vertex after the augmentation (Step 2).

Observation 1 *M_H is a* maximum *matching of H and M_H covers all but at most three vertices: one in each of E_X, E_Y, and E_Z.*



(a) Original co-bipartite graph G. Edges in M^* are strong.

(b) Transformed co-tripartite graph H. Edges in M_H appears in gray.

Fig. 6: Transformation of a co-bipartite graph G with a perfect matching M^*(left) into a co-tripartite graph H (right) on the vertex set M^*.

Lemma 7 *Let u and v be vertices of H that are uncovered by M_H. Then, H does not contain an edge between the partition of u and the partition of v.*

Proof Towards a contradiction, assume that there is an edge xy between the partition of u and the partition of v, such that x is in the partition of u and y is in the partition of v. Then, $xy \notin M_H$, as otherwise $uxyv$ is an augmenting 3-path, contradicting optimality of M_H. Further, $x \neq u$, as otherwise there is an augmenting path involving uy and some vertices of the partition of v. By Observation 1, M_H pairs x with some z and, since M_H is nice, z is in the same partition as x and u. Thus, $uzxy$ is an alternating path from u to y in H ending with xy. By symmetry, there is an alternating path from v to x in H ending with xy. Thus, there is also an alternating path from u to v in H, contradicting optimality of M_H. □

Observation 2 *If $(G, M^*, \sigma_\mathrm{p}, \sigma_\mathrm{c})$ is a yes-instance, then $|V(G)| \geq 4\sigma_\mathrm{c} + 2\sigma_\mathrm{p}$.*

Note that M_H corresponds to a set of alternating 4-cycles in G and, by Observation 1, at most three edges of M^* are not covered by these 4-cycles: one in each of E_X, E_Y and E_Z.

Lemma 8 *Let $(G, M^*, \sigma_\mathrm{p}, \sigma_\mathrm{c})$ be a yes-instance and let $\sigma_\mathrm{c} > |M_H|$. Then, $\sigma_\mathrm{c} = |M_H| + 1$, $\sigma_\mathrm{p} = 0$ and G contains an alternating 6-cycle intersecting E_X, E_Y and E_Z. Moreover, the result of removing any such 6-cycle from G and M_H can be covered with exactly $\sigma_\mathrm{c} - 1$ alternating 4-cycles.*

Proof Assume that G can be covered by a collection S of $> |M_H|$ alternating cycles and any number of alternating paths. Note that, by optimality of M_H, at least one of the cycles of S contains at least 6 vertices of G. Thus, S covers at least $4|M_H| + 6$ vertices of G. But by Observation 1, we have $|V(G)| \leq 4|M_H| + 6$, implying that S contains no paths, $|M_H|$ 4-cycles, and one 6-cycle C of G. Furthermore, M_H covers all but exactly three vertices of H.

Assume that C does not intersect E_Y. Then, $|E_Y|$ is even since $S \setminus C$ is a collection of 4-cycles. But, since M_H covers all but one edge of E_Y, it covers an odd number of edges in E_Y. But then, some edge in E_Y is matched with an edge of E_X or E_Z by M_H, contradicting Lemma 7. Assume that C does not intersect E_X. Then, since C intersects E_Y, all but one vertex of C are in E_Z, implying that G has an alternating 4-cycle intersecting E_Y and E_Z, contradicting Lemma 7.

Finally, we show that the result of removing any 6-cycle C' intersecting E_X, E_Y and E_Z can be covered by $\sigma_\mathrm{c} - 1$ alternating cycles. To this end, it suffices to observe that, by Lemma 7, $|E_X \setminus C'|$, $|E_Y \setminus C'|$ and $|E_Z \setminus C'|$ are all even and each induces a clique in H. □

Since we can find a 6-cycle as described in Lemma 8 in linear time, we can solve unweighted Strict Scaffolding in linear time if $\sigma_\mathrm{c} > |M_H|$. Thus, in the following, we assume $\sigma_\mathrm{c} \leq |M_H|$. Furthermore, if $X \neq \varnothing \neq Z$, $\sigma_\mathrm{p} + \sigma_\mathrm{c} = 1$ and there are no edges between X and Z in G, then the instance is clearly a no-instance, so let us assume that this is not the case. Finally, assume that $|V| \geq 10$, as otherwise, the instance has constant size and can thus be solved in constant time.

Definition 4 An instance of Strict Scaffolding is called *non-trivial* if

– $|V| \geq 10$,
– $|V| \geq 2\sigma_{\mathrm{p}} + 4\sigma_{\mathrm{c}}$,
– $\sigma_{\mathrm{c}} \leq |M_H|$, and
– if $X \neq \varnothing \neq Z$ and $\sigma_{\mathrm{p}} + \sigma_{\mathrm{c}} = 1$, then there are edges between X and Z.

Observation 3 *Let $|V| \geq 10$. Then $E_Y \neq \varnothing$ or $|E_X| \geq 3$ or $|E_Z| \geq 3$.*

Lemma 9 *Every non-trivial instance $(G, M^*, \sigma_{\mathrm{p}}, \sigma_{\mathrm{c}})$ of unweighted* Strict Scaffolding *with $\sigma_{\mathrm{p}} \geq 1$ is a yes-instance.*

Proof Assume that $I := (G, M^*, \sigma_{\mathrm{p}}, \sigma_{\mathrm{c}})$ is a no-instance.

Case 1: $E_Y \neq \varnothing$. If there is some $e \in E_Y$ that is uncovered by M_H, then we can clearly cover M^* by σ_{c} alternating 4-cycles (a subset of the 4-cycles implied by M_H) and one alternating path containing e. Since $|V| \geq 2\sigma_{\mathrm{p}} + 4\sigma_{\mathrm{c}}$, this path can be split into σ_{p} alternating paths. Thus, suppose that all $e \in E_Y$ are covered by M_H and $|M_H| = \sigma_{\mathrm{c}}$ (as otherwise, we can just remove a matching edge of M_H intersecting E_Y). Since $|V| \geq 2\sigma_{\mathrm{p}} + 4|M_H|$ but I is a no-instance, we have $\sigma_{\mathrm{p}} = 1$ and both E_X and E_Y have a vertex that is not covered by M_H. However, since $E_Y \neq \varnothing$, there is a 4-cycle C of G implied by M_H that intersects E_Y and C contains a non-matching edge f in $\binom{X}{2}$ or $\binom{Z}{2}$. By symmetry let $f \in \binom{X}{2}$. But then, C can be extended to a 6-cycle by replacing f with an alternating 3-path using the uncovered edge of E_X and the uncovered edge in E_Z can be covered by one alternating path, thus covering M^* with one path and σ_{c} cycles, contradicting I being a no-instance.

Case 2: $E_Y = \varnothing$. Then, by Observation 3 and symmetry, we can suppose that $|E_X| \geq 3$. If none of the 4-cycles in G that are implied by M_H contains a non-matching edge of $\binom{X}{2}$, then M_H matches at least two edges of E_X with edges of E_Z. Then, H contains an alternating 4-cycle with respect to M_H that can be flipped. Thus, suppose that there is a 4-cycle C_X in G implied by M_H that has a non-matching edge in $\binom{X}{2}$. Then, C_X can be extended or a new path can be used to cover any uncovered edges of M^* in X. If the same holds for $\binom{Z}{2}$, then it is not hard to see that M^* can be covered by σ_{p} paths and σ_{c}. Thus, suppose that there is no 4-cycle in G implied by M_H that has a non-matching edge in $\binom{Z}{2}$. Since $E_Y = \varnothing$ and by the exchange-argument above, $|Z| = 2$ and $|E_Z| = 1$. But then, E_Z can be covered by one path. If $\sigma_{\mathrm{p}} > 1$ or $\sigma_{\mathrm{c}} \geq 1$, then E_X can be covered by $\sigma_{\mathrm{p}} - 1$ paths and σ_{c} cycles simply by adding 1-paths and extending the 4-cycles implied by M_H. Otherwise, $\sigma_{\mathrm{p}} = 1$ and $\sigma_{\mathrm{c}} = 0$ and, by Definition 4, there is an edge between X and Z, allowing the one path that covers E_Z to be extended to also cover E_X. □

We are left with the case that $\sigma_{\mathrm{p}} = 0$ and $\sigma_{\mathrm{c}} \leq |M_H|$, which is slightly more complex. To this end, let us modify M_H slightly by flipping any alternating path in H starting in an $e \in E_Y$ that is unmatched by M_H and ending in E_X or E_Z. By maximality of M_H there is at most one such $e \in E_Y$ and, thus, this operation can be executed in $O(n+m)$ time. Moreover, it does not change the size of M_H. The modification to M_H implies the following observations.

Observation 1 *Let E_Y contain an edge that is not matched by M_H. Then, $|E_Y|$ is odd and, for each non-matching edge e between X and Z, either both or none of the two edges in M^* that are adjacent with e are in E_Y.*

Finally, we characterize which of the remaining non-trivial instances are negative.

Lemma 10 *Let $I := (G = (X \uplus Y, E), M^*, 0, \sigma_c)$ be a non-trivial instance of unweighted* Strict Scaffolding. *Then, I is a no-instance if and only if*

1. *all edges between X and Z are in M^* and $|E_Y|$ is odd,*
2. *$|E_X| = 1$ and no alternating cycle in G covers E_X (or the same for E_Z),*
3. *$\sigma_c = 1$ and no alternating cycle covering E_X in G can cover E_Z (or vice versa), or*
4. *$\sigma_c = |M_H|$, M_H touches all edges of M^* except exactly one $e \in E_Y$, no non-matching edge between X and Z is adjacent with any edge of E_Y, and there is no edge between E_X and E_Z in H.*

Proof "⇒": To show the contraposition, suppose that none of the conditions holds. We construct a solution for I based on M_H. To this end, let S denote the union of all alternating 4-cycles in G that are implied by M_H. Consider the graph J whose vertices are the cycles of S and each C and C' are adjacent if there is a 4-cycle C^* of non-matching edges in G that intersects both $E(C)$ and $E(C')$. Then C^* is alternating with respect to S and flipping it will make S contain one alternating cycle less than before. This operation corresponds to contracting the edge $\{C, C'\}$ in J. Thus, if $M^* \subseteq S$ and the connected components in J are at most σ_c, then a solution can be reached this way. Otherwise, we consider the following cases (noting that, whenever some e is matched to some f by M_H, we know that S contains an alternating cycle in G that contains both e and f).

Case 1: $\exists\, e \in M^* \setminus S$. Note that, by construction of S, we know that e is not matched by M_H.

Case 1a: $e \in E_Y$. Then, by Observation 1, $|E_Y|$ is odd and, by Condition 1, there is some $f \in E \setminus M^*$ between X and Y in G. If f is adjacent with e then, by Observation 1, f is adjacent with another edge $e' \in E_Y$. Observation 1 also implies that M_H matches e' to some $h \in E_Y$ and we can replace e' by the alternating 3-path (e', f, e) in S. The case that f is not adjacent with e, but with another edge e' of E_Y is symmetrical, as e' is matched to some h by M_H and we can just flip the alternating path (e, e', h) to have f adjacent with the unmatched edge in E_Y. In the following, we thus suppose that no edge between X and Z is adjacent with any edge of E_Y and we let $h_x \in E_X$ and $h_z \in E_Z$ be the edges of M^* that f is adjacent with. Let M'_H be the result of repeatedly flipping any alternating path from any unmatched (by M_H) $h \notin \{h_x, h_z\}$ to h_x or h_z in H and note that $|M'_H| = |M_H|$. If h_x or h_z is unmatched by M'_H then let $M''_H = M'_H$. Otherwise, by Condition 4, there is an alternating h_x-h_z-path in G and we let M''_H be the result of flipping this alternating path and removing any pairings that touch h_x or h_z. Then, $|M''_H| \geq |M'_H| - 1 = |M_H| - 1$. Consider the result S' of collecting all 4-cycles implied by M''_H, adding the 6-cycle corresponding to $\{e, h_x, h_z\}$, and extending this 6-cycle to contain all unmatched (by M''_H) edges of G. Then, S covers M^* with at least $|M''_H| + 1 \geq |M_H| = \sigma_c$ alternating cycles and can thus be turned into a solution.

Case 1b: $e \in E_X \cup E_Z$. By symmetry let $e \in E_X$. Then, S does not contain edges of $\binom{X}{2} \setminus M^*$, as those could be replaced by an alternating 3-path containing e. If E_X contains another edge $f \neq e$, then M_H matches f to some edge $h \in E_Z$. If E_X contains a third edge $f' \notin \{e, f\}$, then, by the same argument, M_H matches f' to some edge $h' \in E_Z$ and H contains an alternating 4-cycle (f, h, h', f') that can be flipped in order to let M_H match f with an edge of E_X and the above argument applies. Thus, we can suppose that $E_X = \{e, f\}$. Since we can flip (e, f, h), the same argument applies to E_Z and we conclude $|E_Z| \leq 2$. Then, by Observation 3, E_Y is not empty, but contains an edge p that is matched to some q by M_H (Case 1a treats the case that p is unmatched). If $q \in E_Y \cup E_Z$, then we can flip (e, f, h) and merge h into the alternating cycle of S containing p and q. Thus, $q \in E_X$, implying $q = f$, which contradicts f being matched with h. Therefore, we suppose $E_X = \{e\}$ in the following (implying that S does not intersect $\binom{X}{2}$ at all). By Condition 2, there are alternating cycles in G that cover e. Let C be such an alternating cycle that, among the ones minimizing $|C|$, maximizes $|C \cap E_Y|$. If $|C| = 4$, then there is an edge $\{e, f\}$ in H corresponding to C and f is matched to some $h \neq e$ by M_H. If $f \in E_Y$, then $h \in E_Z$, as S does not contain edges of $\binom{X}{2} \setminus M^*$. If E_Y contains another edge $f' \neq f$, then M_H matches f' with either an edge of E_Y, in which case S contains an edge of $\binom{X}{2} \setminus M^*$, or an edge of E_Z, in which case we can flip the alternating 4-cycle containing f, h, and f' in H to make S contain an edge of $\binom{X}{2} \setminus M^*$. Thus, $E_Y = \{f\}$ and, since I is non-trivial, $|E_Z| \geq 3$. But then, we can flip the alternating path (e, f, h) in H and swap E_X and E_Z, obtaining an instance with $|E_X| = 3$, which was handled previously. Hence, we suppose $|C| \geq 6$ in the following. Note that, by minimality of C, we have either $|C \cap E_Y| = 2$, implying $C \cap E_Z = \varnothing$ or $|C \cap E_Z| = 2$, implying $C \cap E_Y = \varnothing$ or $|C \cap E_Y| = |C \cap E_Z| = 1$ (note that $|C| = 6$ in all cases). Let $C \cap M^* = \{e, f, f'\}$ and let h and h' be matched to f and f', respectively, by M_H. Suppose that f, f', h, and h' are distinct, since otherwise, $\{f, f'\} \in M_H$ and we can just extend the 4-cycle containing f and f' in S to cover e. Note that neither h nor h' are in E_Z as otherwise, either $f, f' \in E_Y$ and S intersects $\binom{X}{2}$ or $|C \cap E_Y|$ is not maximal. But then, we can replace in S the alternating 4-cycles corresponding to $\{f, h\}$ and $\{f', h'\}$ with the alternating 4-cycle corresponding to $\{h, h'\}$ and the alternating 6-cycle C, keeping a total of $|M_H|$ alternating cycles in S that now cover e.

Case 2: $M^* \subseteq S$, but J has more than σ_c connected components. Note that J has at most 3 connected components: one corresponding to 4-cycles avoiding E_Z, one corresponding to 4-cycles avoiding E_X, and one corresponding to 4-cycles intersecting both E_X and E_Z. Indeed, if there are 4-cycles avoiding both E_X and E_Z, then the first two components are joined.

Case 2a: $\sigma_c = 2$. Then, J has three components and no alternating 4-cycle in S avoids both E_X and E_Z. Suppose $E_Y \neq \varnothing$, as otherwise, it is trivial to cover E_X with one alternating cycle and E_Y with another one. Let C be an alternating 4-cycle intersecting both E_X and E_Z and let C' be an alternating 4-cycles intersecting E_Y. Since C' does not avoid both E_X and E_Z, suppose by symmetry that C' intersects E_X. Then, the result S' of merging C and

C' in S contains at least σ_c alternating cycles, covers M^*, and contains an alternating cycle intersecting both E_X and E_Z. Thus, S' can be turned into a solution.

Case 2b: $\sigma_c = 1$. If $E_Y = \varnothing$, then Condition 3 implies that the instance is a yes-instance. If $|E_Y| \geq 2$, then we can construct an alternating cycle C by first connecting E_Y in an alternating cycle intersecting both $\binom{X}{2}$ and $\binom{Z}{2}$ and then replacing any edge in $C \cap \binom{X}{2}$ by an alternating path covering E_X and analogous for E_Z. Note that C covers all but ($|E_Y|$ mod 2) edges of M^*. Thus, suppose in the following that $|E_Y|$ is odd. As $M^* \subseteq S$, there is some $e = xz$ in E_Y (say $x \in X$, $z \in Z$) that is matched with some $f \in E_X$ (or E_Z) by M_H. Let I' be the instance resulting from removing the vertices of e from G and removing e from M^*. If $|E_Y| \geq 3$ or one of E_X and E_Z is empty, then the construction above gives an alternating cycle C covering all edges in $M^* - e$. Let $h = uv$ be a non-matching edge adjacent with f in C such that $u \in f$ is adjacent with both x and z in G. Also, $u \in X$, since $f \in E_X$. If $v \in X$, then we can replace h by (u, y, x, v) in C and, if $v \in Z$, then we can replace h by (u, x, y, v) in C. Thus, in the following, suppose $E_X \neq \varnothing \neq E_Z$ and $|E_Y| < 3$ (that is, $E_Y = \{e\}$). If G contains a non-matching edge h between X and Z that is not adjacent to e and let $h_x \in E_X$ and $h_z \in E_Z$ be adjacent to h, then (h_x, e, h_z) implies an alternating 6-cycle C in G that intersects both $\binom{X}{2} \setminus M^*$ and $\binom{Z}{2} \setminus M^*$ and we can thus extend C to cover E_X and E_Z and, thereby, M^*. Hence, suppose that all non-matching edges between X and Z in G are adjacent to e. But then, no alternating cycle containing e in G can contain any other edge between X and Z, contradicting the assumption that Condition 3 does not hold.

"$\Leftarrow$": As I being a no-instance clearly follows from either of the first three conditions, it remains to show that it also follows from Condition 4. To this end, let Condition 4 hold and assume towards a contradiction that there is a collection S of $|M_H|$ alternating cycles covering M^*. First, since M_H is a matching and e is the only unmatched edge in G, we know that S consists of $|M_H| - 1$ 4-cycles and one 6-cycle C. Second, since no non-matching edge between X and Z is adjacent with any edge of E_Y, no edge of E_Y can form an alternating 4-cycle with any edge in $E_X \cup E_Z$. As H also does not contain edges between E_X and E_Z, we conclude that H is a disjoint union of three cliques: $H[E_X]$, $H[E_Y]$ and $H[E_Z]$. Third, as e is the only unmatched edge, $|E_X|$ and $|E_Z|$ are even and $|E_Y|$ is odd. Thus, not all edges of E_Y can be covered by alternating 4-cycles in G, implying that C contains an edge of E_Y. Since there are no alternating 6-cycles containing three edges of E_Y in G, we conclude that C contains exactly one edge of E_Y. As there are no non-matching edges between X and Z adjacent to this edge, C also contains exactly one edge of E_X and one edge of E_Z. But then, $S \setminus C$ covers an odd number of edges of E_X, contradicting that $S \setminus C$ is a collection of 4-cycles or that H consists of three cliques. □

For all but Condition 3, it is easy to see how to apply them in linear time. Indeed, no alternating cycle covering E_X can cover E_Z if and only if no

alternating cycle intersects E_X and E_Z or there is exactly one such alternating cycle and it has length 4. This concludes the proof of Theorem 9.

4.2 Solving Sparse Graphs

We show that, if G is a quasi forest and $\sigma_{\mathrm{p}} = 0$, then Scaffolding, and even Strict Scaffolding, can be solved in linear time. To this end, we employ the following reduction rule.

Rule 1 *Let u be a leaf in $G - M^*$ such that the parent v of u in $G - M^*$ is not a leaf. Then, delete all edges incident with v in $G - M^*$ that are not uv.*

Proof (Correctness of Rule 1) The proof is based on the argument that any solution S for G is a perfect matching (that is, $S \cup M^*$ has no degree-1 vertices). Since uv is the only edge of $G - M^*$ incident with u, it is apparent that $uv \in S$ and, thus, no other edge incident with v is in S. □

If we maintain a list of leaves on each edge-deletion, we can apply Rule 1 exhaustively in linear time. Moreover, if it is no longer applicable to $G - M^*$, then $G - M^*$ is a matching and checking whether G has the correct number of cycles can be done in linear time. Finally, we can extend this idea to work for any σ_{p} and σ_{c} by guessing all $2\sigma_{\mathrm{p}}$ end points of paths in the solution and deleting the non-matching edges incident with them. Clearly, the result of this operation remains a quasi-forest and all vertices having a parent in $G - M^*$ have degree two in the solution, so the correctness of Rule 1 remains valid.

Corollary 9 Strict Scaffolding *can be solved in $O(n^{2\sigma_{\mathrm{p}}+1})$ time on quasi forests.*

4.3 Polynomial Approximation Algorithms

Unfortunately, Theorem 9 holds only for unweighted instances. As we have seen in 3.2, Scaffolding is $\mathcal{NP}$-hard if we allow weights to be 0 or 1. However, we can still show a simple factor-2 approximation, that is, Algorithm 1 produces a solution of weight at least half the optimum weight, for Max Scaffolding in case G is a complete graph or a complete bipartite graph.

Algorithm 1 starts with a maximal-cardinality maximum-weight matching S of $G - M^*$, implying that $S \cup M^*$ is a collection of cycles. Then, it merges cycles, two at a time. Finally, it turns cycles into paths until the correct numbers of paths and cycles are reached.

Lemma 11 *If G is a complete graph, Algorithm 1 produces a solution whose weight is at least half the optimum.*

Proof Let S^{org} denote the set S as computed in Line 1 and let $\tilde{S}$ denote the set S returned in Line 14. First, we show that $\tilde{S}$ is a solution. To this end, note that S^{org} is a matching in $G - M^*$ and $S^{\mathrm{org}} \cup M^*$ is a collection of cycles

Algorithm 1: A 2-approximation for Max Scaffolding on complete bipartite graphs.

```
1  S ← a maximal-cardinality maximum-weight matching in G − M*;
2  C ← the set of cycles in S ∪ M*;
3  X ← ⋃_{C∈C} argmin{ω(uv) | uv ∈ C \ M*};
4  while |X| > σ_c + σ_p do
5  |   e, e' ← argmin{ω(e), ω(e') | e, e' ∈ X ∧ e ≠ e'};
6  |   Y ← a maximum-weight 4-cycle containing e and e' in G;
7  |   S ← SΔY;
8  |   e* ← argmin{ω(e*) | e* ∈ S ∩ Y};
9  |   X ← (X \ {e, e'}) + e*;
10 while |X| > σ_c do
11 |   e ← argmin{ω(e) | e ∈ X};
12 |   S ← S − e;
13 |   X ← X − e;
14 return S;
```

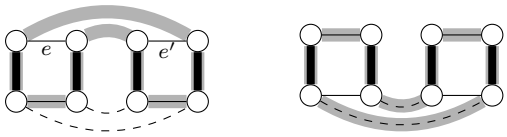


Fig. 7: An example with $\sigma_c = 1$ for which Algorithm 1 gives a solution of half optimal weight. Drawn edges (solid and dashed) have weight 1, all other edges have weight 0. The solid edges are a maximal-cardinality maximum-weight matching. Left: Algorithm 1 replaces e and e' to form the highlighted solution of weight 2. Right: an optimal solution of weight 4.

since S^{org} is maximal-cardinality (and, thus, perfect). Since the only times S changes is when its symmetric difference with a 4-cycles is formed (Line 7) or when edges are removed from S (Line 12), the set $\tilde{S}$ is a matching in $G - M^*$. Thus, $\tilde{S} \cup M^*$ has maximum degree two. Further, note that "$X \subseteq S$" and "$S \cup M^*$ is a collection of cycles" are invariants of the first while loop. Since, in Line 9, we know that $S \cup M^*$ has at most $\sigma_p + \sigma_c$ connected components, all of which are cycles, we conclude that $\tilde{S} \cup M^*$ is a collection of at most σ_p paths and at most σ_c cycles.

Next, we show that the weight of the set S returned in Line 14 is at least half the weight of a maximum matching in $G - M^*$, which is an upper bound on the solution weight and which is equal to $\omega(S^{\text{org}})$. To this end, note that for all cycles C of $S^{\text{org}} \cup M^*$, we selected a minimum-weight edge e_C of C into X in Line 3. Thus, $\omega(C) \geq |C|/2 \cdot \omega(e_C)$ for each cycle C in $S^{\text{org}} \cup M^*$. Finally, let X^{org} denote the set X as computed in Line 3. Then, since $|C| \geq 4$ for each C,

$$\omega(X^{\text{org}}) = \omega(\bigcup_C e_C) \leq \sum_C \omega(e_C) \leq \sum_C 2\omega(C)/|C| \leq \sum_C \omega(C)/2 \leq \frac{1}{2}\omega(S^{\text{org}}).$$

Since Algorithm 1 never touches any edge of S^{org} except edges in X^{org}, we know that $S^{\text{org}} \subseteq \tilde{S} \cup X^{\text{org}}$ and, thus, $\omega(\tilde{S}) \geq \omega(S^{\text{org}}) - \omega(X^{\text{org}}) \geq \omega(S^{\text{org}})/2$. □

Note that all arguments remain valid for complete bipartite graphs. Furthermore, Figure 7 gives an example of a configuration in which Algorithm 1 gives a solution of weight half the optimum, implying that the bound of two is tight.

Theorem 10 *If G is a complete bipartite graph or a clique, then* Max Scaffolding *can be approximated to within a factor 2 in asymptotically the same time as it takes to compute a bipartite matching in G (currently $O(|V|^3)$). This factor is tight.*

5 Variants of Scaffolding

When considering a weighted graph, we studied the problem of identifying a subset of edges whose removal from the graph causes the largest cost increase. This problem is denoted as k most vital edges problem. A dual problem consists of determining a set of edges of minimum cardinality whose removal causes the cost of solution to become greater than a given threshold. This problem is denoted by min edge blocker problem. Those problems have been studied for various classes of combinatorial problems in [2, 3, 4]. The underlying idea is that, for Scaffolding, it may be related to the quest of a "core partial solution", on which we may have a greater confidence, since it has the most impact on the score of an optimal solution. This partial solution may be extended further into a complete solution, by exhaustive exact search for instance. Unfortunately, the problem is already difficult for constrained cases.

5.1 About Vital Edges

Here is the formal definition of the problem, adapted to Scaffolding.

Most vital edges of Scaffolding (Scaffolding(MV))
Input: $G, \omega : E \to \mathbb{N}$, a perfect matching M^* in G, and $\sigma_{\text{p}}, \sigma_{\text{c}}, k, l \in \mathbb{N}$
Question: Is there a size-k set $E' \subseteq E \setminus M^*$ such that $G - E'$ can be covered by a collection S' of $\leq \sigma_{\text{p}}$ alternating paths and $\leq \sigma_{\text{c}}$ alternating cycles and G can be covered by a collection S of $\leq \sigma_{\text{p}}$ alternating paths and $\leq \sigma_{\text{c}}$ alternating cycles such that $\omega(S) - \omega(S') \geq l$?

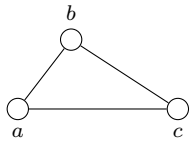
Note that there is a version of Scaffolding(MV) looking to minimize $\omega(S) - \omega(S')$ instead of maximizing it and this version is called *Least vital edges of* Scaffolding. We first consider cases where the lengths of the cycles and paths are fixed by $\ell_{\text{p}} \geq 1$ and $\ell_{\text{c}} = 6$. To show that Strict Scaffolding remains $\mathcal{NP}$-hard under these conditions, we reduce from the following $\mathcal{NP}$-complete problem [16] (see Figure 8).

Partition Into Triangles (PT)
Input: $G = (V, E)$, with $|V| = 3q = n, q \in \mathbb{N}$ and $|E| = m$.
Question: Can the vertices of G be partitioned into q disjoints sets containing exactly three vertices, $T_1, T_2, \ldots, T_q$, such that for each $T_i = \{u_i, v_i, w_i\}, i \in \{1, \ldots, q\}$, all three edges $\{u_i, v_i\}$, $\{u_i, w_i\}$, $\{w_i, v_i\}$ belong to E?

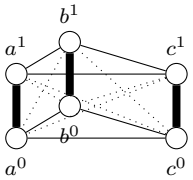
Construction 5 *Let $G = (V, E)$ be an instance of* Partition into Triangles. *We consider the graph $G' = (V' = V_0 \cup V_1, E' = E_0 \cup E_1 \cup E_2 \cup E_3)$:*

- *We consider two copies of G denoted by $G_0 = (V_0, E_0)$ and $G_1 = (V_1, E_1)$ with vertices respectively denoted by x^0 and x^1 for $x \in V$.*
- *$\forall x \in V$, $\{x^0, x^1\} \in E_2$.*
- *$\forall \{x, y\} \in E$, $\{x^0, y^1\} \in E_3$ and $\{x^1, y^0\} \in E_3$.*

The perfect matching M^* consists in the edges of E_2. We also add the following weights on edges outside M^*: $\omega(e) = M, e \in E_0 \cup E_1$, otherwise $\omega(e) = M'$ with $M' < M$. We set $k = 2m$ with m is the number of edges in the graph G.



(a) Original instance of Partition Into Triangles



(b) Transformed instance of Strict Scaffolding with $\ell_p \geq 1$ and $\ell_c = 6$. Edges are solid (weight-M), bold (M^*), or dotted (weight-M').

Fig. 8: Illustration of Construction 5.

Theorem 11 Strict Scaffolding(MV) *is $\mathcal{NP}$-complete, even if $\sigma_p = 0$ and $\ell_c = 6$.*

Proof The problem is clearly in $\mathcal{NP}$. Let $G = (V, E)$ be an instance of Partition Into Triangles, with $n = 3q$ vertices and m edges. We consider the instance $I' = (G', M^*, \omega, (\sigma_c, 12), (0, \ell_p))$. The number of vertices and edges in G' is $2n$ and $4m + n$, respectively. Then, G admits a partition into triangles if and only if there are two solutions S and S' for I' such that the gap between S and S' is $2q(M - M')$ *i.e.* $\omega(S) - \omega(S') = 2(M - M')q$.

- Suppose there is a positive solution for I' such that $\omega(S) - \omega(S') = 2(M - M')q$. Clearly, in S at most two edges with weight M in each triangle may be chosen. So $\omega(S) \leq (2M + 4M')q$. Moreover, $\omega(S') = 6M'q$. Notice that $\omega(S)$ is equal to $(2M + 4M')$ if two edges of weight M are included in each cycle of length six, and the solution S uses the cycle $\{x^0, y^0, y^1, z^0, z^1, x^1, x^0\}$ whereas S' uses the cycle $\{x^0, x^1, y^0, y^1, z^0, z^1, x^0\}$. The union of corresponding triangles in G, $\cup\{x, y, z\}$ is a G-cover.

- Conversely, suppose that G admits a partition into triangles. We construct a feasible solution for I' as follows.
  1. The value of $\omega(S) = (2M + 4M')q$ iff G admits a partition into triangle. For a triangle $\{x, y, z\}$, we consider the following alternating-cycle of length six: $\{x^0, y^0, y^1, z^0, z^1, x^1,\ x^0\}$. It is clear that all alternating-cycles cover the vertices of G'.
  2. For any another solution $S' \neq S$, we have $\omega(S') \geq 6qM'$. Indeed it is sufficient to consider the following alternating-cycle of length six: $\{x^0, x^1, y^0, y^1, z^0, z^1, x^0\}$.

  Therefore, we have $\omega(S) - \omega(S') = 2(M - M')q$.

□

The previous result can be extended to the bipartite case, with $\ell_c = 12$.

Corollary 10 Strict Scaffolding(MV) *with* $\ell_c = 12$ *is* $\mathcal{NP}$*-complete, even on bipartite graphs with maximum degree four.*

Proof The proof is very similar to the previous one, and is based on the slightly different Construction 6, which construct a bipartite graph, where the bound on degree is same as in the original instance of Partition Into Triangles. Notice that Partition Into Triangles remains $\mathcal{NP}$-complete even for maximum degree at most four, yielding this part of the result. This construction is illustrated by Figure 9.

Construction 6 *Let* $G = (V, E)$ *be an instance of* Partition Into Triangles. *We consider the graph* $G' = (V' = V_0 \cup V_1, E' = E_0 \cup E_1 \cup E_2)$*:*

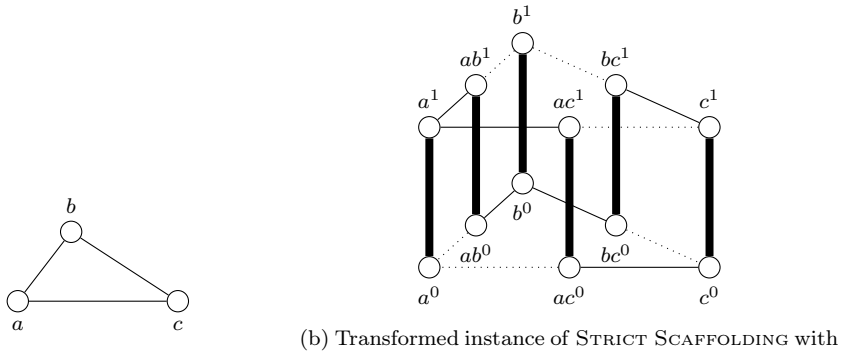
- *We consider two copies of* G *denoted by* $G_0 = (V_0, E_0)$ *and* $G_1 = (V_1, E_1)$ *with vertices respectively denoted by* x^0 *and* x^1 *for* $x \in V$*.*
- *For each edge* $e \in E_0 \cup E_1$*,* e *is split into two edges by adding a new vertex i.e.* $\forall \{x^0, y^0\} \in E_0$ *(resp.* $\forall \{x^1, y^1\} \in E_1$*), we add* x^0y^0 *(resp.* x^1y^1*) and two new edges* $\{x^0, x^0y^0\}$ *and* $\{x^0y^0, y^0\}$ *(resp.* $\{x^1, x^1y^1\}$ *and* $\{x^1y^1, y^1\}$*). The set of vertices* V_i*, and edges* $E_i, i \in \{0, 1\}$ *are updated.*
- *We add all the edges of the form* $\{x^0, x^1\}$ *and* $\{x^0y^0, x^1y^1\}$ *to the set of edges denoted* E_2*.*

The perfect matching consists in the edges of E_2*. We set the following weights: every edge* e *in* G' *is incident to a vertex* xy^0 *or* xy^1*. If* $\omega(x^0xy^0) = M$*, we set* $\omega(xy^1y^1) = M$ *and* $\omega(xy^0y^0) = \omega(x^1xy^1) = M'$ *with* $M' < M$*. We let* $k = 2m$ *where* m *is the number of edges in the graph* G*.*

Clearly by construction the graph G' is bipartite. It is sufficient to consider $l = 2q(M - M')$.

□

Construction 7 *Let* $G = (V, A)$ *an instance of the* Directed Hamiltonian Path *problem. We construct the following graph* $G' = (V', E)$*, obtained from* G *by Construction 1 and add cycles* $(y_j^1, y_j^2, y_j^3, y_j^4), \forall j \in \{1, \ldots, K\}$*, add arbitrary edges* $\{\{v_i^1 y_j^1\}, \forall j \in \{1, \ldots, K\}$ *to* E*, add edges* $\{y_j^1, y_j^2\}$ *and* $\{y_j^3, y_j^4\}$ *in* M^**,* $\forall j \in \{1, \ldots, K\}$*.*



(a) Original instance of Partition Into Triangles

(b) Transformed instance of Strict Scaffolding with $\ell_p \geq 1$ and $\ell_c = 12$. Edges are solid (weight-M), bold (M^*), or dotted (weight-M').

Fig. 9: Illustration of Construction 6.

Corollary 11 Strict Scaffolding(MV) *with* $\ell_c = 4$ *and* $\sigma_p = 1$ *remains* $\mathcal{NP}$*-complete on planar bipartite graphs.*

Proof The proof is based on Construction 7. As previously, there is a positive solution for Directed Hamiltonian Path if and only if there are two solutions S and S' of Strict Scaffolding(MV) such that $\omega(S) - \omega(S') = 2(M - M')\sigma_c$. □

Notice that all previous results may be extended to the problem of k Least vital edges.

Corollary 12 *The* minimization *problem for* Strict Scaffolding(MV) *is non-approximable for all previous problems.*

The following problem is close to previous ones, but aims to minimize the size of the removed edge set.

Min / Max Edge Blocker Strict Scaffolding (Min / Max-Strict Scaffolding(EB))
Input: G, $\omega : E \to \mathbb{N}$, perfect matching M^* in G, $\sigma_p \in \mathbb{N}$, $\sigma_c \in \mathbb{N}$, $k \in \mathbb{N}$.
Question: A subset $E' \subseteq E$ of minimum cardinality with $G - E'$ have a σ_p-σ_c-cover S' with respect to M^* such that $\omega(S')$ is at least / most k i.e. $\omega(S') \geq k$ / $\omega(S') \leq k$?

Both Scaffolding(MV) and Min-Strict Scaffolding(EB) are polynomial-time equivalent.

Corollary 13 Scaffolding(EB) *with* $\sigma_p = 0$ *is* $\mathcal{NP}$*-complete for bipartite graphs.*

Proof It is sufficient to consider the E'-edges set of the solution given by S in the proof of Theorem 11 and put $k = 6qM'$. □

5.2 About Disjoint Solutions

In the following, we consider the problem concerning the existence of two disjoints solutions. Two solutions S_1 and S_2 are edge-disjoint (disjoint in the following) according to a perfect matching M^* if $(S_1 \backslash M^*) \cap (S_2 \backslash M^*) = \varnothing$.

Two Disjoint Solutions for Scaffolding (2-Scaffolding)
Input: $G = (V, E)$, $\omega : E \to \mathbb{N}$, perfect matching M^* in G, $\sigma_{\mathrm{p}}, \sigma_{\mathrm{c}}, k \in \mathbb{N}$
Question: Is there an $S_1, S_2 \subseteq E \backslash M^*$, two disjoint solutions such that $S_i \cup M^*$, for $i = 1, 2$ is a collection of $\leq \sigma_{\mathrm{p}}$ paths and $\leq \sigma_{\mathrm{c}}$ cycles and $\omega(S_i) \geq k$?

We consider the following polynomial-time construction from Directed Hamiltonian Path, illustrated by Figure 10. Notice that the produced graph G' is planar if G is planar.

Construction 8 *Let $G = (V, A)$ be an instance of* Directed Hamiltonian Path. *We construct the graph $G' = (V_0 \cup V_1, E_0 \cup E_1)$ as follows.*

- *For each $u \in V$, we construct a path $\mathcal{P}_{6,u} = (u_1, u_2, u_3, u_4, u_5, u_6)$ and add the two edges (u_2, u_4) and (u_3, u_5). We denote the resulting set of edges as E_0.*
- *For each arc $(u, v) \in A$, we construct a graph $\mathcal{PE}_{u,v}$ with two vertices $(u, v)_0$ and $(u, v)_1$, and add the edges $\{u_6, (u, v)_0\}$, $\{(u, v)_0, (u, v)_1\}$, $\{(u, v)_1, v_1\}$, $\{u_6, (u, v)_1\}$ and $\{(u, v)_0, v_1\}$. Such vertices are in V_1 and the corresponding edges in E_1.*

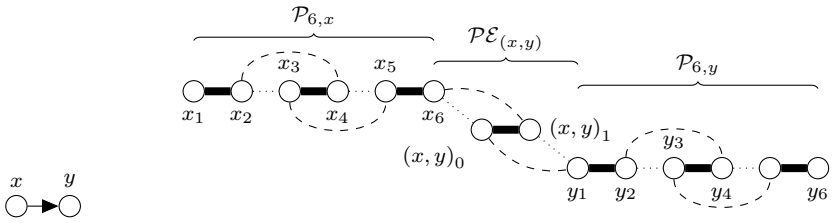
We construct the perfect matching M^* on G', consisting of the edges of the kind $\{u_1, u_2\}, \{u_3, u_4\}, \{u_5, u_6\}, \forall u \in V$ and $\{(u, v)_0, (u, v)_1\}, \forall (u, v) \in A$.

Theorem 12 *2-Scaffolding with $\sigma_{\mathrm{c}} = 0$ is $\mathcal{NP}$-complete, even on planar graphs. Moreover, assuming the Exponential-Time Hypothesis, there is no $2^{o(n)}$-time algorithm in this case.*

Proof Clearly, there are two disjoint paths according to the perfect matching M^* for the path $\mathcal{P}_{6,u}$ *i.e.* $u_1 \to u_2 \to u_3 \to u_4 \to u_5 \to u_6$ (denoted by $\bar{\mathcal{P}}_{6,u}$) and $u_1 \to u_2 \to u_4 \to u_3 \to u_5 \to u_6$ (denoted by $\widetilde{\mathcal{P}_{6,u}}$). Similarly, for an edge-path of length three $\mathcal{PE}_{(u,v)}$, we may consider the two disjoint paths $u_6 \to (u, v)_0 \to (u, v)_1 \to v_1$ (denoted by $\bar{\mathcal{PE}}_{(u,v)}$) or $u_6 \to (u, v)_1 \to (u, v)_0 \to v_1$ (denoted by $\widetilde{\mathcal{PE}_{(u,v)}}$).

Therefore according to the previous discussion, it is clear that there is a solution for Directed Hamiltonian Path if and only if there is a solution for 2-Scaffolding, *i.e.* two disjoint solutions S_1 and S_2. Indeed, each solution S_i uses the $\bar{\mathcal{P}}_{6,u}$ or $\widetilde{\mathcal{P}_{6,u}}$ and $\bar{\mathcal{PE}}_{(u,v)}$ or $\widetilde{\mathcal{PE}_{(u,v)}}$ paths.

Moreover, the previous polynomial-time transformation is linear which implies the results for subexponential-time algorithms. □



(a) Original arc. (b) Transformed instance of the arc xy. Matching edges are bold.

Fig. 10: Example of Construction 8.

6 Conclusion

In this article, we presented an overview of the negative and positive results in terms of complexity and approximation for Scaffolding. Refining previously obtained results, we were particularly interested in different classes of graphs, some of them because they have a resemblance to real scaffold graphs, particularly due to their sparsity, and others because we hope to generalize the results of complexity and approximation obtained in complete graphs to these "almost complete" graphs. Negative results concern strong restrictions on the problem, including the number of cycles, paths, their length, the maximum degree of the graph, and gives little hope of finding a polynomial case whose configuration looks like a real graph. In addition, for dense graphs, but also for quasi-forests, we prove several $\mathcal{NP}$-completeness results concerning the optimization problem as soon as we allow two different weights on the edges of the graph. We complement these results with lower bounds on the complexity of exact algorithms for these problems under the Exponential-Time-Hypothesis. Negative results for approximation are also exposed, especially for the minimization problem, even in quasi-forests and graphs containing bipartite graphs. Continuing the quest for effective angles of attack for the problem, we studied parameterized algorithms, in particular, we excluded polynomial kernels with respect to the parameter treewidth for which the problem is known to be $\mathcal{FPT}$.

However, we also found promising positive results, in particular for dense graphs. We proved that the decision problem is polynomial for co-bipartite graphs, and exhibited an approximation algorithm with a factor of 2 for scaffolding in the cliques and bipartite complete graphs.

Finally, we have shown that it is equally difficult to find a subset of "vital" edges for Scaffolding, as the problem itself.

These results raise interesting new questions which are to be explored, aiming to approach the boundaries of the problem. For instance, there is a difference in the complexity of the decision problem in co-bipartite graphs and split graphs, which we consider structurally quite close. From these results, we also wish to infer approximation algorithms with a performance guarantee for Scaffolding in these graph classes, for example by adapting a greedy strategy or using a maximum perfect matching like in the case of cliques. If

these results are confirmed, we hope to extend them to classes of graphs that generalize the concept of cliques and independent sets. A promising candidate concept is that of (r, l)-graphs, that is, graphs whose vertices can be partitioned into r independent sets and l cliques [8].

Our results for approximation in complete graphs require a series of tests on real and simulated dataset, to examine if the ratio obtained in practice would be better than 2. It is expected indeed that it is not, since actual scaffold graphs are rather sparse and many edges will be of weight zero. On a theoretical level, one wonders if this approximation algorithm can be generalized to a $\mathcal{PTAS}$.

As for parameterized algorithms, we can look closer to other parameters, or search for parameterized approximation algorithms that would be a first step towards a practical tackling of the problem. It was also the underlying idea of considering the Most Vital edges problems. These issues would deserve a little extra exploration, particularly on sparse graphs.

Acknowledgements This work was supported by the Institut de Biologie Computationnelle[2] (ANR Projet Investissements d'Avenir en bioinformatique IBC).

References

1. Alimonti P, Ausiello G, Giovaniello L, Protasi M (1997) On the complexity of approximating weighted satisfiability problems. Tech. rep., Università degli Studi di Roma La Sapienza, rapporto Tecnico RAP 38.97
2. Bazgan C, Toubaline S, Vanderpooten D (2012) Critical edges/nodes for the minimum spanning tree problem: complexity and approximation. Journal of Combinatorial Optimization 26(1):178–189
3. Bazgan C, Bentz C, Picouleau C, Ries B (2015) Blockers for the stability number and the chromatic number. Graphs and Combinatorics 31(1):73–90
4. Bazgan C, Nichterlein A, Niedermeier R (2015) A refined complexity analysis of finding the most vital edges for undirected shortest paths. In: Paschos VT, Widmayer P (eds) Algorithms and Complexity - 9th International Conference, CIAC 2015, Paris, France, May 20-22, 2015. Proceedings, Springer, Lecture Notes in Computer Science, vol 9079, pp 47–60
5. Bodlaender HL (1993) A tourist guide through treewidth. Acta Cybernetica 11(1-2):1–21
6. Bodlaender HL (2016) Treewidth of graphs. In: Encyclopedia of Algorithms, pp 2255–2257
7. Bodlaender HL, Jansen BMP, Kratsch S (2014) Kernelization lower bounds by cross-composition. SIAM J Discrete Math 28(1):277–305
8. Brandstädt A (1996) Partitions of graphs into one or two independent sets and cliques. Discrete Mathematics 152(1–3):47 – 54
9. Chateau A, Giroudeau R (2014) Complexity and Polynomial-Time Approximation Algorithms around the Scaffolding Problem. In: Proc. AlCoB '14, Springer, LNCS, vol 8542, pp 47–58
10. Chateau A, Giroudeau R (2015) A complexity and approximation framework for the maximization scaffolding problem. Theoretical Computer Science 595:92 – 106
11. Chen J, Chor B, Fellows M, Huang X, Juedes DW, Kanj IA, Xia G (2005) Tight lower bounds for certain parameterized NP-hard problems. Inf Comput 201(2):216–231
12. Crescenzi P (1997) A short guide to approximation preserving reductions. In: Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997, pp 262–273

[2] http://www.ibc-montpellier.fr/

13. Dayarian A, Michael T, Sengupta A (2010) SOPRA: Scaffolding algorithm for paired reads via statistical optimization. BMC Bioinformatics 11:345
14. Downey RG, Fellows MR (2013) Fundamentals of Parameterized Complexity. Texts in Computer Science, Springer
15. Gao S, Sung WK, Nagarajan N (2011) Opera: Reconstructing Optimal Genomic Scaffolds with High-Throughput Paired-End Sequences. Journal of Computational Biology 18(11):1681–1691
16. Garey MR, Johnson DS (1979) Computers and Intractability: A Guide to the Theory of NP-Completeness. publisher-freeman
17. Håstad J (1997) Clique is hard to approximate within $n^{1-\varepsilon}$. Electronic Colloquium on Computational Complexity (ECCC) 4(38)
18. Hunt M, Newbold C, Berriman M, Otto T (2014) A comprehensive evaluation of assembly scaffolding tools. Genome Biology 15(3):R42
19. Impagliazzo R, Paturi R (2001) On the Complexity of k-SAT. Journal of Computer and System Sciences 62(2):367–375
20. Impagliazzo R, Paturi R, Zane F (2001) Which problems have strongly exponential complexity? J Comput Syst Sci 63(4):512–530
21. Lokshtanov D, Marx D, Saurabh S (2011) Lower bounds based on the exponential time hypothesis. Bulletin of the EATCS 105:41–72
22. Plesník J (1979) The NP-Completeness of the Hamiltonian Cycle Problem in Planar Digraphs with Degree Bound Two. Inf Process Lett 8(4):199–201
23. Weller M, Chateau A, Giroudeau R (2015) Exact approaches for scaffolding. BMC Bioinformatics 16(Suppl 14):S2
24. Weller M, Chateau A, Giroudeau R (2015) On the complexity of scaffolding problems: From cliques to sparse graphs. In: Lu Z, Kim D, Wu W, Li W, Du D (eds) Combinatorial Optimization and Applications - 9th International Conference, COCOA 2015, Houston, TX, USA, December 18-20, 2015, Proceedings, Springer, Lecture Notes in Computer Science, vol 9486, pp 409–423
25. Woeginger G (2003) Exact algorithms for np-hard problems: A survey. In: Combinatorial Optimization — Eureka, You Shrink!, Lecture Notes in Computer Science, vol 2570, Springer Berlin Heidelberg, pp 185–207