Scaffolding Problems Revisited: Complexity, Approximation and Fixed Parameter Tractable Algorithms, and Some Special Cases

Mathias Weller, Annie Chateau, Clément Dallard, Rodolphe Giroudeau

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

Mathias Weller[1], Annie Chateau[2,3], Clément Dallard[4], and Rodolphe Giroudeau[3]

[1]CNRS, LIGM, Marne-la-Vallée, France
[2]IBC, Montpellier, France
[3]LIRMM, Montpellier, France
[4]University of Portsmouth, UK

July 19, 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.

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 [8]. Some methods use heuristic ways to simplify the graph [13], others use a decomposition of the problem into two separate steps (orienting and ordering), whose difficulty could be bypassed under certain restrictions [11]. A good presentation of the mainly used recent methods can be found in [16].

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. Edges in the matching represent the contigs, whereas 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 [7, 8]. The beginnings of these results are presented in [21]. 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 [22, 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" [22], 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 2-SAT 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 an fixed-parameter tractable algorithm. Section 3.3 describes how cross-composition leads to a negative result in quasi-forests.

G	$\omega_{\max}=1$			$\omega_{\max}=0$	
	$\sigma_p, \sigma_c > 0$	$\sigma_c = 0$	$\sigma_p = 0$	$\sigma_p > 0$	$\sigma_p = 0$
bipartite	$\mathcal{NP}$c (Theorem 2 & [8])				
co-bipartite	$\mathcal{NP}$c (Corollary 2), no $2^{o(m)}$-time algorithm (Corollary 3)			$\mathcal{P}$ (Theorem 8)	
split				open	
quasi forest	$\mathcal{NP}$c, $\mathcal{W}[1]$-h wrt. k (Thm. 4) no $2^{o(m)}$- or $n^{o(k)}$-time algorithm (Corollary 5)		$\mathcal{P}$ (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 9)	$\notin \mathcal{APX}$ (Cor. 6 & [8]), $\omega_{\max}/\omega_{\min}$-approximation
co-bipartite, split	open	
quasi forest	no $n^{\frac{1}{2}-\epsilon}$-approx. (Cor. 7)	$\mathcal{APX}$-hard (Cor. 8)

Table 2: Complexity to approximate Scaffolding.

We consider also dense graphs who we know are susceptible to polynomial-time approximation algorithms [7, 8]. 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 [8] 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 and Table 2. Next section is devoted to formal description of problems.

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 this depressing review, we focus on the hopeful part of the work, in Section 4. Afterwards, we enlarge our point of view by considering several variants of the problem, unfortunately all $\mathcal{NP}$-complete, in Section 5.

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 $G-V' := G[V \setminus V']$. Further, for any $S \subseteq E$, we define $\mathrm{Gr}(S) := (\bigcup_{e\in S} e, S)$ and $G-S := (V, E\setminus S)$. 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 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, $\sigma_p, \sigma_c, k \in \mathbb{N}$
Question: Is there an $S \subseteq E \setminus M^*$ such that $\mathrm{Gr}(S \cup M^*)$ is a collection of $\leq \sigma_p$ alternating paths and $\leq \sigma_c$ 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 *exactly* σ_p paths and *exactly* σ_c cycles is called Strict Scaffolding (SSCA). When we want to precise particular values for σ_p and σ_c, we refer to the problem as $(\sigma_p, \sigma_c)-$Scaffolding. If we are looking for paths and cycles of *fixed lengths* ℓ_p and ℓ_c, we replace σ_p and

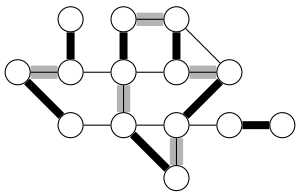


Figure 1: An example of instance of Scaffolding. Matching edges are strong. With $(\sigma_p, \sigma_c) = (2, 2)$, it is positive for Scaffolding, but negative for Strict Scaffolding. A solution is given in gray.

σ_c by pairs (σ_p, ℓ_p) and (σ_c, ℓ_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. A quasi-forest is *linear* if it is a collection of paths.

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* [17, 18] 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 $(x, k) \in Q$ 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 When it is hard

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 Triangle, or Weighted 2-SAT.

In the following paragraphs, we use a reduction from the Directed Hamiltonian Path (resp. Directed Hamiltonian Cycle) ([14]) 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 / DNC)

Input: A directed graph G without self loop
Question: Does G contain a simple path visiting all vertices?

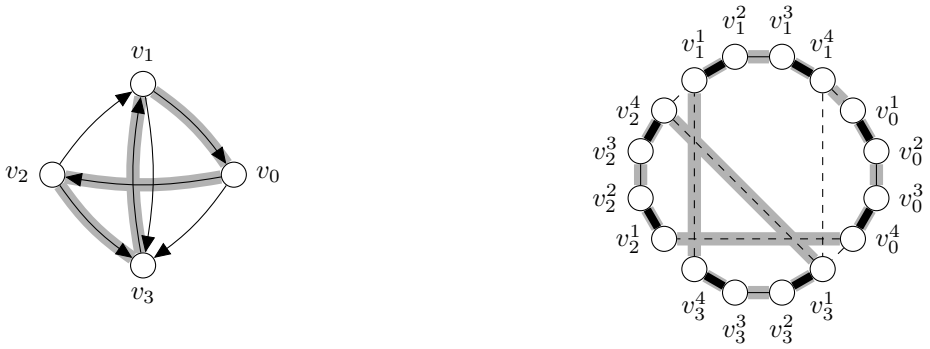


Figure 2: Example of Construction 1, transforming the left instance of Directed Hamiltonian Cycle to the right graph with edges of M^* in bold and edges of the form $v_i^4 v_j^1$ dashed. A corresponding solution is highlighted.

Construction 1. *Let $G = (V = \{v_1, v_2, \ldots, v_n\}, A)$ be an instance of DHC. We construct $G' = (V_1 \uplus V_2, E)$ as follows (see Figure 2).*

$$V_1 := \{v_i^1, v_i^3 \mid v_i \in V\} \qquad V_2 := \{v_i^2, v_i^4 \mid v_i \in V\}$$

$$E := \{v_i^1 v_i^2, v_i^2 v_i^3, v_i^3 v_i^4 \mid v_i \in V\} \cup \{v_i^4 v_j^1 \mid v_i v_j \in A\}.$$

Finally, let $M^* := \{v_i^1 v_i^2, v_i^3 v_i^4 \mid v_i \in V\}$, let $\omega : E \to \{0\}$, and let $k := 0$.

3.1 To solve

3.1.1 Decision problem with strong constraints on (σ_p,ℓ_p) and (σ_c,ℓ_c)

In [8], we proved that, when the number of edges in the matching is equal to $\sigma_p + 2\sigma_c$, the $(\sigma_p, \sigma_c)-$Strict Scaffolding problem is polynomial, because it forces cycles to 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 $(\sigma_p, \sigma_c)-$Strict Scaffolding (decision problem) in planar bipartite graphs in presence of one path and cycles of length four. We show that the problem remains $\mathcal{NP}$-complete. The reduction relies on the following polynomial-time transformation, based on Construction 1. Notice that DHP remains $\mathcal{NP}$-complete for planar directed maximum-degree-3 graphs ([20]). Let K be an arbitrary constant.

Construction 2. *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 $\{\{u_1 y_j^1\}, \forall j \in \{1, \ldots, K\}$ to E, add edges $\{y_j^1, y_j^2\}$ and $\{y_j^3, y_j^4\}$ in M^*, $\forall j \in \{1, \ldots, K\}$*

Theorem 1. *The problem $((\sigma_p, \ell_p), (\sigma_c, \ell_c))-$Strict Scaffolding with parameters $(\sigma_p, \ell_p) = (1, |V| - 4K - 1)$ and $(\sigma_c, \ell_c) = (K, 4))$ is $\mathcal{NP}$-complete, even if the graph is a planar cubic bipartite graph. Assuming the Exponential-Time Hypothesis, there exists no hope to find a algorithm in $O(2^{o(\sqrt{n})})$ time for the $((\sigma_p, \ell_p), (\sigma_c, \ell_c))-$Strict Scaffolding in presence of a bipartite planar graphs even with cycle of length four.*

Proof. Since in the transformed instance there is no other cycles of length four than the added ones, it is clear that it exists a positive solution for Directed Hamiltonian Path if and only if it exists a positive solution for $((\sigma_p, \ell_p), (\sigma_c, \ell_c))-$Strict Scaffolding with $(\sigma_p, \ell_p) = (1, |V| - 4K - 1)$ and $(\sigma_c, \ell_c) = (K, 4))$. Moreover, the previous polynomial-time transformation is linear which implies the results for subexponential-time algorithm. □

Corollary 1. *There is no hope to find a polynomial-time approximation algorithm within a ratio $\rho < 5/4$ (resp. a $\mathcal{XP}$-algorithm for* Scaffolding *parameterized by number of cycles).*

Again, using Construction 1, we can show that Scaffolding is $\mathcal{NP}$-complete on a very restricted class of graphs.

Theorem 2. *Unweighted* Scaffolding *is $\mathcal{NP}$-complete on bipartite planar graph with maximum degree three, even if $\sigma_p = 0$ and $\sigma_c = 1$.*

Proof. The problem is clearly in $\mathcal{NP}$. We show that it is also $\mathcal{NP}$-hard by proving that G has a Hamiltonian cycle if and only if G' has an alternating Hamiltonian cycle with respect to M^*.

"⇒": Let $\mathcal{C}$ be a Hamiltonian cycle in G. Then, $S := \{v_i^2 v_i^3 \mid v_i \in V\} \cup \{v_i^4 v_j^1 \mid v_i v_j \in \mathcal{C}\}$ is a feasible solution, and $\mathrm{Gr}(S \cup M^*)$ is an alternating Hamiltonian cycle with respect to M^*.

"⇐": Let S' be a matching in $G' - M^*$ such that $\mathcal{C}' := \mathrm{Gr}(S \cup M^*)$ is an alternating Hamiltonian cycle in G' with respect to M^*. Since $\mathcal{C}'$ contains $v_i^3 v_i^4$ 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 $v_i^4 v_j^1$ for $v_i v_j \in A$. Since S' is a matching in $G - M^*$, no two of these edges are adjacent. Now, $\mathcal{C} := \{v_i v_j \mid v_i^4 v_j^1 \in S'\}$ 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.

By construction, G' is a bipartite planar graph with Max-Degree 3 according to Transformation 1 since Directed Hamiltonian Cycle remains $\mathcal{NP}$-complete for planar graph with Max-Degree 3 (see [20]). □

3.1.2 Weighted cases in dense graphs

Note that we can change Construction 1 such that $\omega : E \to \{1\}$ and $k := 4n$. This further enables us to add any number of edges of weight 0 without affecting the correctness argument. This implies that Scaffolding is $\mathcal{NP}$-complete on, for example, split graphs and co-bipartite graphs.

Corollary 2. *Let $\mathfrak{G}$ be a class of graphs such that, for each bipartite graph G there is a supergraph of G in $\mathfrak{G}$.* Scaffolding *is $\mathcal{NP}$-complete on $\mathfrak{G}$, even if $\sigma_{\mathrm{p}} = 0$ and $\sigma_{\mathrm{c}} = 1$ 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 [18, 24]). In fact, the lower bound is established directly from the fact that (planar) Directed Hamiltonian Cycle does not admit a $O(2^{o(|E(G)|)})$-time algorithm [19, Theorem 3.5] and that Construction 1 only linearly blows up the instance size.

Corollary 3. *Let $\mathfrak{G}$ be a class of graphs such that, for each bipartite graph G there is a supergraph of G in $\mathfrak{G}$. Assuming ETH, there is no $2^{o(|E(G)|)}$-time algorithm for* Scaffolding *on $\mathfrak{G}$, even if $\sigma_{\mathrm{p}} = 0$ and $\sigma_{\mathrm{c}} = 1$ and $\omega_{\max} = 1$.*

3.1.3 Weighted cases in sparse graphs

The hardness of Scaffolding for dense graphs proved by Theorem 2 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) [22], 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 2-SAT to Scaffolding, that does not require fixing the lengths.

Construction 3. *Let $G = (V, A)$ be an instance of* Directed Hamiltonian Cycle*, with $V = \{v_1, v_2, \ldots, v_n\}$. We construct $G' = (V', E)$ from G as follows:*

- *For each $u \in V$, we construct a "vertex-path" $P_{4,u} = (u_1, u_2, u_3, u_4)$, and we call $\{u_2, u_3\}$ an* inner edge*. The set of all such paths is denoted by $P_4 = \bigcup_{u \in V} P_{4,u}$*
- *For each $(u, v) \in A$, we construct an "edge-path" with four vertices, denoted $PE_{4,uv} = (uv^1, uv^2, uv^3, uv^4)$ and the two edges $\{u_4, uv^1\}, \{uv^4, v\}$.*
- *We add a path $Z = (z_1, z_2, z_3, z_4)$ plus the edges $\{z_1, z_3\}$ and $\{z_2, z_4\}$.*
- *For each vertex $u \in V$, we add the edges $\{u_1, z_1\}, \{u_2, z_1\}, \{u_4, z_1\}$. For each $(u, v) \in A$, we add the edge $\{uv^2, z_1\}$.*
- $M^* := \{\{u_1, u_2\}, \{u_3, u_4\} \mid u \in V\} \cup \{\{uv^1, uv^2\}, \{uv^3, uv^4\} \mid (u, v) \in A\} \cup \{\{z_1, z_2\}, \{z_3, z_4\}\}$.

Lemma 1. *$G' - M^*$ is a tree.*

Proof. First, $G' - M^*$ is a connected graph since all vertices, except z_1 are connected to z_1. To show that $G' - M^*$ has no cycles, we count the number of edges in $G' - M^*$. For each $v \in V$, we have the edges $\{z_1, v_1\}$, $\{v_2, v_3\}$, $\{z_1, v_2\}$, and $\{z_1, v_4\}$, and for each $(u, v) \in A$ we have the edges $\{u_4, uv^1\}$, $\{z_1, uv^2\}$, $\{z_1, uv^3\}$, $\{uv^4, v_1\}$. Finally, we have the edges $\{z_1, z_3\}, \{z_2, z_4\}, \{z_2, z_3\}$. Therefore, $|E| = 4n + 4|A| + 3 = |V'| - 1$. So $G' - M^*$ is a tree. □

Theorem 3. Scaffolding *with $(\sigma_{\mathrm{p}}, \ell_{\mathrm{p}}) = (|E| - n + 1, 3)$ and $(\sigma_{\mathrm{c}}, \ell_{\mathrm{c}}) = (1, 8n)$ is $\mathcal{NP}$-complete on quasi-trees.*

[1] The ETH states: there is a constant $c > 1$ such that no $O(c^n)$-time algorithm for n-variable 3-SAT exists.

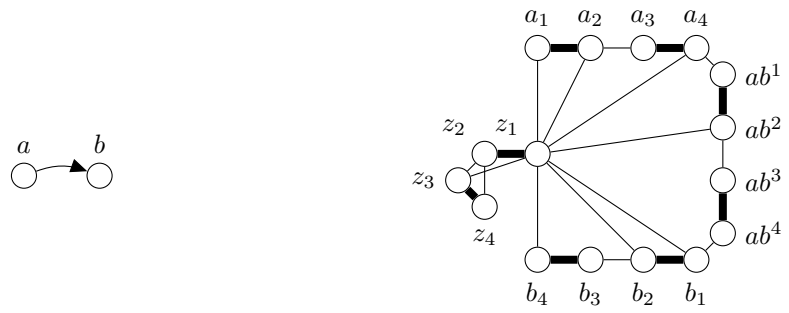
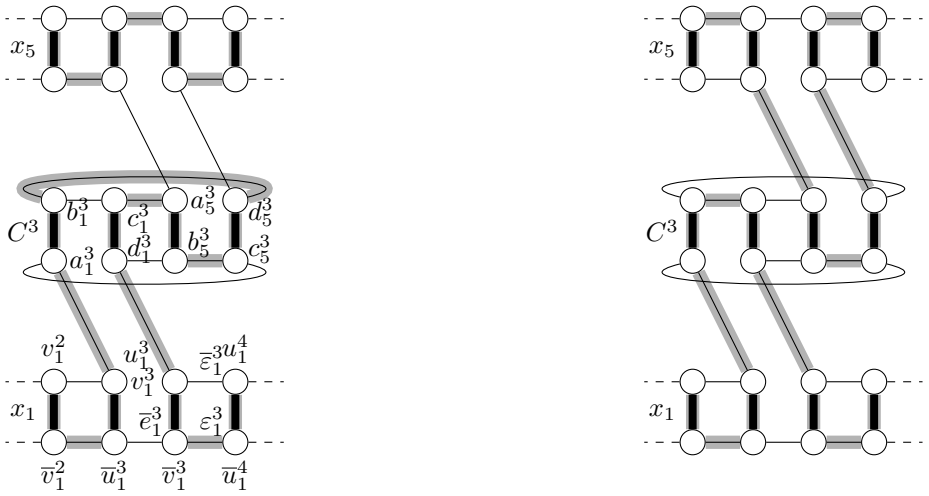


Figure 3: Illustration of Construction 3.



(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 $x_5 = 0$.

Figure 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.

Proof. Clearly, the problem is in $\mathcal{NP}$. We show that Construction 3 is correct, that is, G has a Hamiltonian cycle if and only if (G', M^*) can be covered by $|E| - n + 1$ paths of length 3 and 1 cycle of length $8n$.

"⇒": Let $\mathcal{C} = (v_1, v_2, \ldots, v_n)$ be a directed Hamiltonian cycle in G. We construct a solution for the Scaffolding-instance as follows: The alternating cycle of length $8n$ consists of the vertex-paths P_{4,v_i} for all $i \leq n$ and the edge-paths $PE_{4,uv}$ with $(u, v) \in \mathcal{C}$. A path of length 3 is given by Z, the remaining paths are given by the paths-edges $PE_{4,uv}$ such that $(u, v) \notin \mathcal{C}$.

"⇐": Suppose there is a set of one alternating cycle $\mathcal{C}$ of length $8n$ and $|E| - n + 1$ alternating paths of length 3. Clearly, the vertices of Z are not in $\mathcal{C}$ and therefore, they are included in a path of length 3. Thus, the edges $\{z_1, x\}$ for each $x \in V' \setminus \{z_2, z_3, z_4\}$ cannot be used in the covering. By construction, vertex-paths (resp. edge-paths) cannot appear consequently in $\mathcal{C}$. Since each vertex- and edge- path contains exactly four vertices and $\mathcal{C}$ has $8n$ vertices, $\mathcal{C}$ alternately contains n vertex-paths and n edge-paths. Then, a Hamiltonian cycle in G is given by the order of the vertex-paths in $\mathcal{C}$. □

Note that we had to fix the lengths of the paths and cycles we are looking for in the Scaffolding-instance. To show that Scaffolding is also hard on quasi-forests without restricting the lengths, we give another reduction from the $\mathcal{NP}$-complete Weighted 2-SAT problem (see [1]) to Scaffolding.

Weighted 2-SAT

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 2-SAT that ask to find a satisfying assignment β that minimizes or maximizes $\sum_{i \mid \beta(x_i)=1} w_i$ are called Min W2SAT and Max W2SAT, respectively.

Construction 4. *Let (φ, k) be an instance of* Weighted 2-SAT *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 $0 \leq j \leq m$, introduce*

- *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 2. *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 $\mathrm{Gr}(S \cup M^*)$ 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 $\mathrm{Gr}(S \cup M^*)$, the "or" above are exclusive and all other vertices have degree exactly two in $\mathrm{Gr}(S \cup M^*)$, 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 $\mathrm{Gr}(S \cup M^*)$ 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 $\mathrm{Gr}(S \cup M^*)$. 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 occuring 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. □

Since Weighted 2-SAT 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 [12]), by Lemma 2, so is Scaffolding.

Theorem 4. Scaffolding *is $\mathcal{NP}$-hard and $\mathcal{W}[1]$-hard with respect to k, even on bipartite graphs G with $G - M^*$ being a linear forest, $\omega_{\max} = 1$ and $\sigma_c = 0$.*

Construction 4 can be modified to restrict the problem even further: consider two paths $p = (v_0, v_1, \ldots)$ and $q := (u_0, u_1, \ldots)$ in $G - M^*$. We can add new vertices α_j, β_j, γ_j with $j \in \{u, v\}$ with matching edges $\alpha_u\alpha_v$, $\beta_u\gamma_u$, $\beta_v\gamma_v$ and non-matching edges $\gamma_u\gamma_v$, $v_0\alpha_v$, $u_0\alpha_u$ of weight 0 and non-matching edges $\alpha_u\beta_u$, $\alpha_v\beta_v$ of weight $n + 1$. Finally, we ask for a solution of weight $2n(n + 1) + k$ containing $\sigma_p := 2n$ paths. Then, since all solutions have to contain the 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.

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

In analogy with Corollary 3, Construction 4 implies subexponential-time lower bounds for exact algorithms.

Corollary 5.

- *Assuming ETH, there is no $2^{o(|E(G)|)}$-time algorithm for* Scaffolding*, and,*
- *assuming $\mathcal{W}[1] \neq \mathcal{FPT}$, there is no $n^{o(k)}$-time algorithm for* Scaffolding*,*

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

Proof. 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 2-SAT instance. Then, since Independent Set does not have a $2^{o(m)}$-time algorithm [18] (with m denoting the number of edges), Scaffolding does not have a $2^{o(m)}$-time algorithm (unless the ETH fails).

Furthermore, note that k-Independent Set $\equiv$ k-Weighted 2-SAT and that k-Weighted 2-SAT reduces to k-Scaffolding by Construction 4. Thus, since Independent Set does not have an $n^{o(k)}$-time algorithm [9], Scaffolding does not have an $n^{o(k)}$-time algorithm (unless $\mathcal{W}[1] = \mathcal{FTP}$). □

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. Clearly, the isolated matching edges must constitute isolated paths. Further, if any isolated 4-cycle is covered by two paths, there are only $(n-2)$ paths and a cycle to cover all of the u_i^m, $\overline{u}_i^m$, v_i^{-1}, $\overline{v}_i^{-1}$, which can be seen to be impossible.

3.2 To approximate

Furthermore, we derive inapproximability of Scaffolding from Construction 1.

Corollary 6. *Let $\mathfrak{G}$ be a class of graphs such that, for each bipartite graph G there is a supergraph of G in $\mathfrak{G}$. For all $\rho \in \mathbb{N}$,* 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^*. Then, we let $\omega : E' \to \mathbb{N}$ such that $\omega(E_1) = 0$ 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 3.1.2 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 $\frac{\omega_{\max}}{\omega_{\min}}$ (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 W2SAT is $\mathcal{NP}$-hard to approximate to within a factor of $n^{1-\epsilon}$ for any $\epsilon > 0$ [1, 15] and the number of vertices in the instance produced by Construction 4 is bounded in the number of variables, we conclude that, in contrast to the factor-3 approximation for Scaffolding in complete graphs [8] (and the factor-2 approximation presented in subsubsection 4.1.1), the problem is hard to approximate in the restricted class described above.

Corollary 7. Max Scaffolding *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, $\omega_{\max} = 1$ and $\sigma_c = 0$.*

For the minimization version, Min Scaffolding, we derive approximation hardness as well. To see this, note that Construction 4 is an S-reduction (see [10]) and Min W2SAT 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 [8]. Thus, we suspect that Corollary 8 can be strengthened to at least the same hardness-level as we have for Max Scaffolding (Corollary 7). To this end, we conjecture existence of a gap-preserving reduction from an $\mathcal{NP}$-complete problem Π to Min Scaffolding with a non-constant gap.

3.3 To find a polynomial kernel

In order to rule out polynomial kernels, we will use the recent technique of cross-composition [5]. 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 of size polynomial in maximum size of the input instances and 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 $\mathcal{NP} \subseteq co\mathcal{NP}/poly$[2].

Definition 1 (Polynomial equivalence relation [5])**.** *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) [5])**.** *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 5 ([5])**.** *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 $\mathcal{NP} \subseteq co\mathcal{NP}/poly$.*

We now consider the following problem, which is a variant of Weighted 2-SAT, called *Global Verification*, in which we want to find a solution for a given value k, which is not a solution for $k - 1$:

Weighted 2-SAT(GV)

Input: φ a formula in $2 - CNF$ 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., $\sum_{i \mid \beta(x_i)=1} w_i \leq k$ and such that $(\varphi, \omega, k-1) \notin$ Weighted 2-SAT?

Lemma 3. *The problem* Weighted 2-SAT(GV) *is $\mathcal{NP}$-complete.*

Proof. The proof is based on a reduction from the $\mathcal{NP}$-complete problem Vertex Cover(GV).

Vertex Cover(GV)

Input: $G = (V, E)$ a graph, $k \in \mathbb{N}$, and s.t. $\nexists |VC(G)| \leq k - 1$
Question: Is there a vertex-cover $VC(G) \subseteq V$ s.t. $|VC(G)| \leq k$?

The $\mathcal{NP}$-completeness of Vertex Cover(GV) cames from a reduction from the classic problem $3 - SAT$: let I be a instance of $3 - SAT$, and $G = R(\varphi)$ the graph with $2n + 3m$ vertices, one vertex labelled in positive (resp. negative) form for each variable in $3 - SAT$, and one triangle for one clause and $n + 6m$ edges: $3m$ for triangles, n for the two labelled-joined vertices and $3m$ for the relation between vertices-triangles and labelled vertices such that if x occurs in clause C then it exists a edge (x, y) with $y \in C$-triangle. With this construction Vertex Cover(GV) is $\mathcal{NP}$-complete by considering $k = 2m + n$.

Now, we will prove the $\mathcal{NP}$-completeness of Weighted 2-SAT(GV). Let $G = (V, E)$ be a graph with $n = |V|$ and $m = |E|$ and $k \in \mathbb{N}^*$. An instance of Weighted 2-SAT(GV) is constructed the following way:

[2] co$\mathcal{NP}$/poly is the class of decision problems refutable by a family of polynomial-size Boolean circuits.

$\forall v \in V$ we create a weighted boolean variable $\omega(x_v) = c$. $\forall (u,w) \in E$ we add a clause of length two $(x_u \vee x_w)$. The instance admits n variables and m clauses of length two. Notice that an instance may be designed as monotone Weighted 2-SAT(GV).

So, it is obvious that it $\exists VC(G) \leq k$ and $\nexists VC(G) \leq k-1$ if and only if there is a truth assignment β s.t., $\sum_{i \mid \beta(x_i)=1} \omega(x_i) \geq k \times c$ by considering $VC = \{v \in V | x_v = 1\}$. □

Construction 5. *Let $\mathcal{F} = \{(\varphi_1, k), (\varphi_2, k), ..., (\varphi_t, k)\}$ a set of t equivalent instances of* Weighted 2-SAT(GV) *(according to Definition 1). We use the Γ-transformation defined in Construction 4, in order to obtain the set of graphs $\mathcal{S} = \{S_1, S_2, ..., S_t\}$, with $\forall i \in [1,t], S_i = \Gamma(\varphi_i, k)$. We note $\mathcal{S} = \Lambda(\mathcal{F})$, and $G(\mathcal{S}) = \bigcup_{i=1}^{t} S_i$.*

Lemma 4. *Let $\mathcal{F} = \{(\varphi_1, k), (\varphi_2, k), ..., (\varphi_t, k)\}$ and $\mathcal{S} = \Lambda(\mathcal{F})$.*
Then $(\varphi_i, k) \in$ Weighted 2-SAT(GV) $\forall i \in [1,t]$ iff $G(\mathcal{S}) \in$ Scaffolding with $\sigma_{\mathrm{p}} = n \times t, \sigma_{\mathrm{c}} = 0$ and cost $= k \times t$.

Proof. By definition of Weighted 2-SAT(GV), we know that $\forall i, (\varphi_i, k-1)$ is not satisfiable. Moreover, $(\varphi, k) \in$ Weighted 2-SAT(GV) if and only if $\Gamma(\varphi, k) \in$ Scaffolding with $\sigma_p = n \times t, \sigma_c = 0$ and *cost* $= k$ since Γ is a S-réduction, which is a strict reduction preserving the optimal cost (see Construction 4).

Therefore, we have $\Gamma(\varphi, k-1) \notin$ Scaffolding with $\sigma_p = n \times t, \sigma_c = 0$ and *cost* $= k-1$.

- If $(\varphi_i, k) \in$ Weighted 2-SAT(GV), $\forall i \in [1,t]$, then $\Gamma(\varphi_i) \in$ Scaffolding, $\forall i \in [1,t]$ with $\sigma_p = n, \sigma_c = 0$ $k = k_i$, and obviously $G(\mathcal{S}) \in SCA$ with $\sigma_p = n, \sigma_c = 0, cost = k \times t$.
- If $G(\mathcal{S}) \in$ Scaffolding with $\sigma_p = n \times t, \sigma_c = 0$ and *cost* $= k \times t$: consider costs k_i with $\Gamma(\varphi_i) \in$ Scaffolding with $\sigma_p = n, \sigma_c = 0, cost = k_i$, and $\sum_{i=1}^{t} k_i = k \times t$. Since $(\varphi_i, k-1) \notin$ Weighted 2-SAT(GV), $k_i \geq k$, then $\forall i, k_i = k$ in order to satisfy the sum. □

Theorem 6. Scaffolding *in quasi-trees does not admit a polynomial kernel parameterized by treewidth, unless $\mathcal{NP} \subseteq co\mathcal{NP}/poly$.*

Proof. Let $\mathcal{F} = \{(\varphi_1, k), (\varphi_2, k), ..., (\varphi_t, k)\}$ be a set of t instances of Weighted 2-SAT(GV), equivalent according to Definition 1. Using Construction 4, we obtain the set of graphs $\mathcal{S} = \{S_1, S_2, ..., S_t\}$, with $S_i = \Gamma(\varphi_i), \forall i \in [1,t]$, and the graph $G(\mathcal{S}) = \cup_{i=1}^{t} S_i$. The graph $G(\mathcal{S}$ being the union of disjoint connected components, we have $tw(G(\mathcal{S})) = \max_{i \in [t]} \ tw(S_i)$. The Γ-transformation ensures that for a formula φ_i with n variables and m clauses, then S_i possesses $4(n+m)$ vertices. Since all *treewidth* may be bounded by the number of vertices (minus one), and the size of a formula instance of Weighted 2-SAT(GV) polynomially depends on the number of clauses/variables, we have, for a polynomial $P(\cdot,\cdot)$, $tw(G(\mathcal{S})) \leq P(max_{i \in [t]} \ |\varphi_i|, log\ t)$. □

Definition 3. *A connector is a polynomial operator which connects two disjoints paths p_1 and p_2 of G, such that p_1 et p_2 belong to a same path.*

The connector $\oplus$ is defined for two paths p_1 and p_2 of a quasi-tree, where $P = (p_1 \oplus p_2)$ is a path such that $p_1, p_2 \subseteq P$ and P respects the property of the quasi-tree.

Theorem 7. *There is no hope to find a connector $\oplus$ for* Scaffolding *unless $\mathcal{NP} \subseteq co\mathcal{NP}/poly$.*

Proof. Let $\mathcal{F} = \{(\varphi_1, k), (\varphi_2, k), \ldots, (\varphi_t, k)\}$ a set of t instances of Weighted 2-SAT(GV), equivalent according to Definition 1, and $G(\mathcal{S})$ the graph obtained by Construction 4. Each S_i possesses exactly n paths, since $\forall i \in \{1, \ldots, t\}, \varphi_i$ admits a solution of weight at least k. Thus there are $n \times t$ paths in $G(\mathcal{S}$, and p_i^j designates the jth path of S_i.

Suppose that it exists a connector $\oplus$ which connect two paths respecting the quasi-tree property. Therefore, we may extend the process for all $G(S)$-paths. $G_c(S)$ is the resulting graph, with only one path P containing all paths $p_i^j, \forall i, j$.

Thus $\forall i, (\varphi_i, k) \in$ Weighted 2-SAT(GV) iff $\Gamma(\varphi) \in$ Scaffolding with $\sigma_p = 1, \sigma_p = 0$ and *cost* $= k$. Since all instances in $\mathcal{F}$ are equivalent according to $\mathcal{R}_{kn}$, and $\sigma_{\mathrm{p}} = 1$ is polynomial in $\max_{i=1,\ldots,t} |x_i| + \log_{\sum} t$, Scaffolding does not admit a polynomial kernel in σ_{p} on the quasi-tree, unless $\mathcal{NP} \subseteq co\mathcal{NP}/poly$. Nevertheless, for $\sigma_{\mathrm{p}} = c \in \mathbb{N}$ it exists a polynomial-time algorithm for the Scaffolding in presence in quasi-tree. So there is no hope to find a connector. □

4 There is still hope to find

4.1 Polynomial cases

4.1.1 In dense graphs

To solve Strict Scaffolding (decision) in nearly complete graphs, we use a maximum matching in an auxiliary graph. In the following, let G be a co-bipartite graph and M^* a perfect matching in G. Let H be the graph

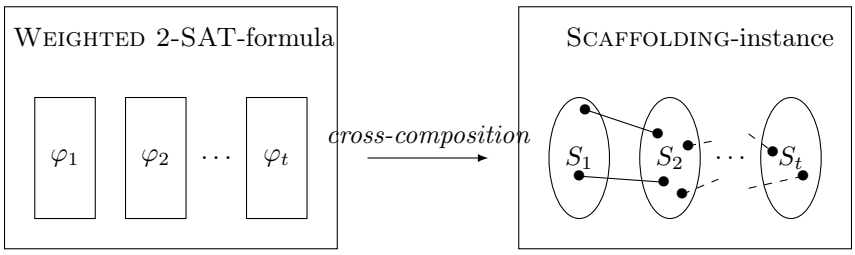
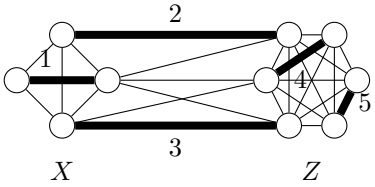
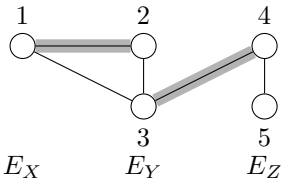


Figure 5: Illustration of the *Cross-composition with a connector.*

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, let M_H be a maximum matching in H.



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



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

Figure 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^*.

Observation 1. *Let H be a co-tripartite graph and let M_H be a maximal matching in H. Then, M_H covers all but at most three vertices: one in each set of the partition.*

Proposition 1. *A maximum matching in a co-tripartite graph can be found in $O(n + m)$ time.*

Sketch. Clearly, a *maximal* matching M'_H in a co-tripartite graph H can be computed in linear time. Further, by Observation 1, at most one vertex in each partition is not covered by M'_H. Then, by the graph structure of H, if M'_H is augmentable, it can be augmented by an easy-to-find augmenting path. □

Lemma 5. *Let H be a co-tripartite graph and let M_H be a maximum matching in H. 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. Further, if a third vertex is uncovered by M_H, then H consists of three disjoint cliques.*

Proof. We suppose that u and v are uncovered by M_H. If there is an edge xy between the partition of u and the partition of v, with x in the partition of u and y in the partition of v, we consider both following cases:

- if the edge xy is in M_H, then $uxyv$ would be an augmenting 3-path, contradicting optimality of M_H.
- if the edge xy is not in M_H: if $u \neq x$, then there is some z matched with x by M_H and, by the first part, z is in the same partition as x and u. Thus, there is an alternating (wrt. M_H) 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.

If a third vertex of H is uncovered by M_H, it implies that all three cliques are disjoint. □

Observation 2. *Let $(G, M^*, \sigma_{\mathrm{p}}, \sigma_{\mathrm{c}})$ be 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 6. *Let $(G, M^*, \sigma_{\mathrm{p}}, \sigma_{\mathrm{c}})$ be a yes-instance, let H and M_H be as defined above 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.

(a) Original graph G.

(b) Graph H. The covering after the covering procedure is in gray.

Figure 7: Case where $\sigma_c \leq |M_H|$ and $\sigma_p \neq 0$ and there are uncovered M^*-edges. Notice that it is a no-instance for $\sigma_p = 1, \sigma_c = 0$, even if $|V(G)| \geq 4\sigma_c + 2\sigma_p$.

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 5. 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 5.

Finally, we show that the result of removing any 6-cycle C' intersecting E_X, E_Y and E_Z can be covered by $\sigma_c - 1$ alternating cycles. To this end, it suffices to observe that, by Lemma 5, $|E_X \setminus C'|$, $|E_Y \setminus C'|$ 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 6 in linear time, we can solve Strict Scaffolding in linear time if $\sigma_c > |M_H|$. Thus, in the following, we assume $\sigma_c \leq |M_H|$.

In a first step, we extend the cycles represented by M_H to cover the edges represented by the uncovered vertices in H. To this end, we first slightly modify M_H:

1. If $u \in E_X$ (or E_Z) is not covered by M_H and H has an alternating (wrt. M_H) path from u to some $z \in E_Y$, then flip this path with respect to M_H.
2. If $u \in E_X$ (or E_Z) is not covered by M_H and H has an alternating (wrt. M_H) path from u to some $z \in E_Z$ not intersecting E_Y, and with an edge between E_X and E_Z not in M_H, then flip this path.
3. If H has an alternating (wrt. M_H) 4-cycle intersecting $\binom{E_Y}{2}$, but not $\binom{E_Y}{2} \cap M_H$, then flip this 4-cycle.

As the result is also a maximum matching in H, we assume in the following that M_H has been modified in this way. Further, these modifications can also be done in linear time.

With this modification, we can discuss the case where $\sigma_p \neq 0$. We define the following *covering* procedure:

1. If $\sigma_c < |M_H|$ and there is an edge in M_H incident with a vertex of E_Y, then remove this edge from M_H.
2. If $\sigma_c < |M_H|$ and no edge in M_H is incident with a vertex of E_Y, arbitrarily remove edges from M_H until reaching σ_c.
   At this step, there is exactly σ_c cycles of length four covered by M_H. We set $\mathcal{C}$ this set of cycles. Since $|V(G)| \geq 4\sigma_c + 2\sigma_p$, we have at least σ_p edges in M^* which are still uncovered.
3. Cover σ_p of these edges of M^* by one path each, beginning with edges in E_Y.
4. If there is some $uv \in E_Y$ that is not covered by M_H, then we cover all edges of $M^* \cap E_Y$ that are not covered by M_H using a single alternating path by extending one initial covered path $uv \in E_Y$.
   At this step, either E_Y is entirely covered, or it is empty and there are still σ_p paths to cover.
5. The following cases occur only if E_Y is empty: If X is totally uncovered, then cover one matching edge in X. If Z is totally uncovered and their remains at least one path available, cover one matching edge in Z.
6. If we have just covered $k < \sigma_p$ distinct paths yet, thus the k covered paths are constituted each by a single matching edge, and there remains sufficiently uncovered matching edge. We cover $\sigma_p - k$ of them arbitrarily.
7. If there is an uncovered edge uv in E_X (resp E_Z), and an extremity of a covered path ending in X (resp. Z), then extend this path with uv.
8. If there is uncovered edge uv in E_X (resp E_Z), and a non-matching edge xy of a covered path or cycle entirely in X (resp. Z), then switch xy with $xuvy$ to cover uv.

After this last step, the only case where there is still uncovered matching edge is when X and Z are disjoint, $\sigma_c = 0$ and $\sigma_p = 1$. Thus, the instance is a no-instance.

In the following, we assume $\sigma_p = 0$. We design the second *covering* procedure as follows: Let $\mathcal{C}$ be initialized as the set of alternating (wrt. M^*) cycles in G that correspond to M_H. We extend $\mathcal{C}$ to cover the edges of M^* that have not been covered by M_H. Next, for each edge $uv \in E_X \cup E_Z$ that is not covered by M_H, find a 3-path $uxyv$ in $G - M^*$ such that xy is in a cycle $C \in \mathcal{C}$. Then, augment C by uv, that is, exchange xy for ux, uv, and vy in C. If, after this, there is at most one edge in $E_X \cup E_Z$ that is not covered by $\mathcal{C}$, then repeat this procedure for an edge $uv \in E_Y$ that is not covered by M_H.

Lemma 7. *Let $\mathcal{C}$ be the result of the above augmentation procedure and let $\sigma_c \leq |M_H|$. If there is at least one edge of M^* that is not covered by $\mathcal{C}$, then the instance is a no-instance.*

Proof. Let S be a solution for $(G, M^*, \sigma_p, |M_H|)$. We consider cases depending on σ_p.

Case 1: $\sigma_p = 0$. Let uv be an edge of M^* that is not covered by $\mathcal{C}$ and let C be a cycle of S containing uv.

(a) Original graph G.

(b) Graph H. The covering after the covering procedure is in gray.

Figure 8: Case where $\sigma_c \leq |M_H|$ and $\sigma_p = 0$ and there are uncovered M^*-edges. Notice that it is a no-instance for $\sigma_p = 0, \sigma_c = 1$, even if $|V(G)| \geq 4\sigma_c + 2\sigma_p$.

- If uv is in X, and at least one other edge in M^* is in X, say xy, then xy is covered by M_H. Let $zt \in M^*$ be a neighbour of xy (wrt M_H). If zt is in E_X, then by the augmenting procedure above, uv would have been covered by M_H. If zt is in E_Y, then by the first modification of M_H, the path $uv-xy-zt$ would have been flipped, and uv covered by M_H. If zt is in E_Z, again by the first modification of M_H, uv would have been covered by M_H. Thus, uv is the lonely edge in E_X (resp. E_Z). Let xy and zt be the neighbours of uv in $M^* \cap C$. We suppose that $ux \in E(G)$ and $vz \in E(G)$. We have to consider the following excluding cases:

  1. $xy = zt$ and this edge is in E_Z. Then E_X is connected to E_Z, and the edge xy is covered by M_H. It neighbour by M_H is either in E_Z, then by the second modification of M_H, the path between xy and this edge would have been flipped, and xy covered by M_H, or it is in E_Y, and then by first modification of M_H, again this path would have been flipped. In any case, xy would have been covered by M_H.
  2. $xy \in E_Z$, $zt \in E_Z$, and $xy \neq zt$. Thus by the augmenting procedure, they are covered by M_H. Indeed, the argument is similar as above. If they belong to a same cycle of $\mathcal{C}$, then the edge xz is such $uxzv \in G \setminus M^*$ and xz is in a cycle. Thus by above augmenting procedure, uv would have been covered by M_H.
  3. $xy \in E_Y$ and $zt \in E_Y$. Then, one of them is covered by M_H, say xy. If its neighbour by M_H is in E_Y, then without loss of generality, we can suppose that it is zt, since this neighbour is then also connected to uv. In this case, by augmenting procedure above, uv would have been covered by M_H. If the neighbour is not in E_Y, then it is in E_Z, and then by the second augmenting procedure, zt would have been included in the cycle. By first augmenting procedure, so does uv.
  4. $xy \in E_Z$, $zt \in E_Y$ (and symmetric case). If none of xy and zt is covered by M_H, it means $|M_H| = 0$, which is not possible since $\sigma_c \geq 1$ and $\sigma_c \leq |M_H|$ by hypothesis. Again, if zt is covered by M_H, by a similar discussion as above, we conclude that uv would have been covered by M_H through one of the augmenting procedure. Suppose now that zt is not covered by M_H, and that xy is covered by M_H. If it neighbour is in E_Y, we can also conclude, since it is also connected to uv. If its neighbour is in E_Z, then by the second augmenting procedure, zt would have been covered by M_H. In any case, it leads to a contradiction.

- The case where $uv \in E_Z$ is totally symmetric.

- If $uv \in E_Y$, we use a very similar discussion. Indeed, xy and zt may be equal or one in E_X and one in E_Z, and we find again a case like above.

In any case, it leads to a contradiction, so uv can not be in a solution and the instance is a no-instance.

Case 2: $\sigma_p \neq 0$. We suppose that there is at least one edge not covered by $\mathcal{C}$. Let uv be an edge of M^* that is not covered by $\mathcal{C}$. We saw above that the only case where there are uncovered edges in M^* is when X and Z are disjoint, $\sigma_c = 0$ and $\sigma_p = 1$, and the instance is a no-instance. □

Theorem 8. *Unweighted* Scaffolding *can be solved in $O(n + m)$ time on co-bipartite graphs.*

4.1.2 In sparse graphs

We show that, if G is a quasi forest and $\sigma_p = 0$, 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.*

Correctness of Rule 1. The proof is based on the argument that any solution S for G is a perfect matching (that is, $\mathrm{Gr}(S \cup M^*)$ 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. □

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 Gr(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;
```

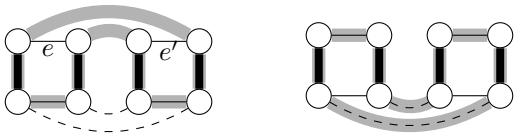


Figure 9: An example with $\sigma_c = 1$ for which Algorithm 1 gives a solution of half optimal weight. Drawn edges (solid and dashed) have weight 1, all other edges have weight 0. The solid edges are a maximal-cardinality maximum-weight matching. Left: Algorithm 1 replaces e and e' to form the highlighted solution of weight 2. Right: an optimal solution of weight 4.

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 $2\sigma_p$ 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* $O(n^{2\sigma_p+1})$ *time on quasi forests.*

4.2 Polynomial approximation algorithms

Unfortunately, Theorem 8 holds only for unweighted instances. As we have seen in 3.1.2, 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 $\mathrm{Gr}(S \cup M^*)$ 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 8. *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 $\mathrm{Gr}(S^{\mathrm{org}} \cup M^*)$ is a collection of cycles 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, $\mathrm{Gr}(\tilde{S} \cup M^*)$ has maximum degree two. Further, note that "$X \subseteq S$" and "$\mathrm{Gr}(S \cup M^*)$ is a collection of cycles" are invariants of the first while loop. Since, in line 9, we know that $\mathrm{Gr}(S \cup M^*)$ has at most $\sigma_p + \sigma_c$ connected components, all of which are cycles, we conclude that $\mathrm{Gr}(\tilde{S} \cup M^*)$ 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 $\omega(S^{\mathrm{org}})$. To this end, note that for all cycles C of $\mathrm{Gr}(S^{\mathrm{org}} \cup M^*)$, 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 $\mathrm{Gr}(S^{\mathrm{org}} \cup M^*)$. 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 9 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 9. *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 Variant problems for 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 a 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 Hardness results for the k Most/Least vital edges Scaffolding

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

> k Most / Least vital edges of Scaffolding (k-MV / LV-Scaffolding)
>
> **Input:** G, $\omega : E \to \mathbb{N}$, perfect matching M^* in G, $\sigma_{\text{p}} \in \mathbb{N}$, $\sigma_{\text{c}} \in \mathbb{N}$, $k \in \mathbb{N}$.
> **Question:** Does it exist a subset $E' \subseteq E, |E'| = k$ with $G - E'$ have a σ_{p}-σ_{c}-cover S' (resp. G have a σ_{p}-σ_{c}-cover S) with respect to M^* such that $(\omega(S) - \omega(S'))$ is $\geq l$ / $\leq l$?

Notice that $E' \cap M^* = \varnothing$.

We first consider cases where the lengths of the cycles and paths are fixed, namely $((\sigma_{\text{p}}, \ell_{\text{p}} \geq 1), (\sigma_{\text{c}}, 6))$-Strict Scaffolding. We now consider a reduction from the Partition Into Triangles [14].

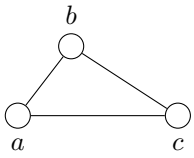
> Partition Into Triangles (PT)
>
> **Input:** $G = (V, E)$, with $|V| = 3q = n, q \in \mathbb{N}$ and $|E| = m$.
> **Question:** Can the vertices of G be partitioned into q disjoints sets containing exactly three vertices, $T_1, T_2, \ldots, T_q$, such that for each $T_i = \{u_i, v_i, w_i\}, i \in \{1,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

We define a polynomial-time transformation from an instance of Partition Into Triangles to an instance of $((\sigma_{\text{p}}, \ell_{\text{p}} \geq 1), (\sigma_{\text{c}}, 6))$-Strict Scaffolding (see Figure 10).

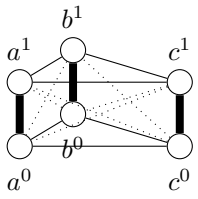
Construction 6. *Let $G = (V, E)$ be an instance of* Partition into Triangles. *We consider the graph $G' = (V' = V_0 \cup V_1, E' = E_0 \cup E_1 \cup E_2 \cup E_3)$:*

- *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 $((\sigma_p, \ell_p \geq 1), (\sigma_c, 6))$-Strict Scaffolding. Edges in $M^*$are strong. Plain edges have weight M, dotted edges have weight M'.

Figure 10: Illustration of Construction 6.

Theorem 10. *The problem k-MV-$((\sigma_p, \ell_p \geq 1), (\sigma_c, 6))$-Strict Scaffolding is $\mathcal{NP}$-complete, even if $\sigma_p = 0$.*

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 graph G' obtained from G by Construction 6. The number of vertices (resp. edges) in G' is $2n$ (resp. $4m + n$). The graph G admits a partition into triangle if and only if there exists two solutions S and S' to $((\sigma_p, \ell_p \geq 1), (\sigma_c, 6))$-Strict Scaffolding such that the gap between S and S' is $2q(M - M')$ *i.e.* $\omega(S) - \omega(S') = 2(M - M')q$.

- Suppose there exists a positive solution for the problem k-MV-$((\sigma_p, \ell_p \geq 1), (\sigma_c, 6))$-Strict Scaffolding 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, we suppose that there exists in G a partition into triangles, let us construct a positive solution for the k-MV-$((\sigma_p, \ell_p \geq 1), (\sigma_c, 6))$Strict Scaffolding in the graph G'.

  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. *The problem k-MV-$((\sigma_p, \ell_p \geq 1), (\sigma_c, 12))$-Strict Scaffolding remains $\mathcal{NP}$-complete for bipartite graphs with bounded degree at most four.*

Proof. The proof is very similar to the previous one, and is based on the slightly different Construction 7, 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 11.

Construction 7. *Let $G = (V, E)$ be an instance of* Partition Into Triangles. *We consider the graph $G' = (V' = V_0 \cup V_1, E' = E_0 \cup E_1 \cup E_2)$:*

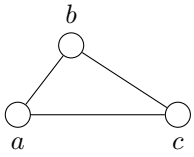
- *We consider two copies of G denoted by $G_0 = (V_0, E_0)$ and $G_1 = (V_1, E_1)$ with vertices respectively denoted by x^0 and x^1 for $x \in V$.*

- *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.*

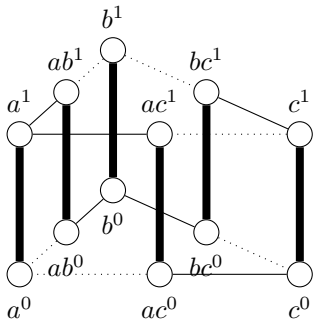
- *We add all the edges of the form $\{x^0, x^1\}$ and $\{x^0y^0, x^1y^1\}$ to the set of edges denoted E_2.*

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 $l = 2q(M - M')$. □



(a) Original instance of Partition Into Triangles



(b) Transformed instance of $((\sigma_p, \ell_p \geq 1), (\sigma_c, 12))$-Strict Scaffolding. Edges in $M^*$are strong. Plain edges have weight M, dotted edges have weight M'.

Figure 11: Illustration of Construction 7.

Corollary 11. *k-MV-Strict Scaffolding remains $\mathcal{NP}$-complete for bipartite planar graph with $\ell_c = 4$ even if $\sigma_c = l \in \mathbb{N}$ and $\sigma_p = 1$ and $\ell_p \geq 1$.*

Proof. The proof is based on Construction 2. We add to this construction, the set of edges $E_2 = \{(y_j^1 y_j^4), (y_j^3 y_j^2)\}, \forall j \in [1, \sigma_c]$ $\omega(e) = M, \forall e \in E_1, \omega(e) = M', \forall e \in E_2$ with $M > M'$, otherwise $\omega(e) = 1$. Finally, we put $k = 2\sigma_c$. As previously, there is a positive solution for Directed Hamiltonian Path if and only if there are two solutions S and S' of k-MV-Strict Scaffolding 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 k-MV-Strict Scaffolding 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 Scaffolding (Min/Max-EB-Scaffolding)

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 problems k-MV-Scaffolding and Min-EB-Scaffolding are polynomial-time equivalent.

Corollary 13. *The problem k-MB-Scaffolding is $\mathcal{NP}$-complete for bipartite graph and the result remains true even if $\sigma_p = 0$.*

Proof. It is sufficient to consider the E'-edges set of the solution given by S in the proof of Theorem 10 and put $k = 6qM'$. □

5.2 Existence of two disjoints 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 $(S_1 \backslash M^*) \cap (S_2 \backslash M^*) = \varnothing$.

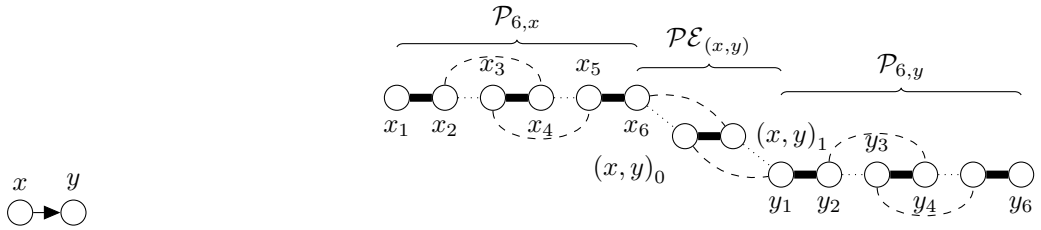
Two Disjoint Solutions for Scaffolding (2-Scaffolding)

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

We consider the following polynomial-time construction from Directed Hamiltonian Path, illustrated by Figure 12. Notice that the produced graph G' is planar if G is planar.

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

- *$\forall u \in V$, we construct a graph with six vertices $\mathcal{P}_{6,u} = (u_1, u_2, u_3, u_4, u_5, u_6)$ with edge between u_i and u_{i+1} for $i = 1, \ldots, 5$. This set of edges are denoted by E_0. Moreover, we add to E_0 these two edges (u_2, u_4) and (u_3, u_5).*



(a) Original arc.

(b) Transformed instance of the arc. Matching edges are strong.

Figure 12: Example of Construction 8.

- $\forall (u,v) \in A$, *we construct a graph* $\mathcal{PE}_{u,v}$ *with two vertices* $(u,v)_0$ *and* $(u,v)_1$, *and add following 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 in 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 11. *The problem 2-*Scaffolding *is* $\mathcal{NP}$*-complete, even if the graph is planar and* $\sigma_c = 0$. *Assuming the Exponential-Time Hypothesis, there exists no hope to find a algorithm in* $O(2^{o(\sqrt{n})})$ *time for the 2-*Scaffolding *in presence of a planar graph.*

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 previous discussion, it is clear that it exists a positive solution for Directed Hamiltonian Path if and only if it exists a positive solution for 2-Scaffolding, *i.e.* two disjoint solutions S_1 and S_2. Indeed, each solution S_i use 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 algorithm. □

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 in these "almost complete" graphs, the results of complexity and approximation obtained in 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, in the side of dense graphs, but also in quasi-forest, we prove several $\mathcal{NP}$-completeness results for 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 a bipartite graph. Continuing the quest for effective angle of attack for the problem, we have also been sought in $\mathcal{FPT}$ algorithms, in particular, we looked if there was a polynomial kernel to the problem parameterized by treewidth. We proved that it could not exist.

However, we also found promising positive results. Regarding 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 have to be explored if we want to approach the boundaries of the problem. Thus, we hope to get a result of polynomiality in split graphs for the decision problem, in a manner similar to that obtained in co-bipartite graphs. 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, it is hoped to extend them in classes of graphs which generalize the concept of cliques and stable, that is (r,l)-graphs, which are graphs which are decomposable into r independent sets and l cliques [6].

Also, results for approximation in complete graphs requires 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 he is, since actual

scaffold graphs are rather sparse and many edges will be of weight zero. On theoretical level, one wonders if this approximation algorithm can be generalized to a $\mathcal{PTAS}$.

As for $\mathcal{FPT}$ algorithms, we can look closer to other parameters, or search for $\mathcal{FPT}$ approximation algorithms that would be a first step towards a practical tackling of the problem. It was also the underlying idea of considering the k-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[3] (ANR Projet Investissements d'Avenir en bioinformatique IBC).

References

[1] P. Alimonti, G. Ausiello, L. Giovaniello, and M. Protasi. On the complexity of approximating weighted satisfiability problems. Technical report, Università degli Studi di Roma La Sapienza, 1997. Rapporto Tecnico RAP 38.97.

[2] C. Bazgan, S. Toubaline, and D. Vanderpooten. Critical edges/nodes for the minimum spanning tree problem: complexity and approximation. *Journal of Combinatorial Optimization*, 26(1):178–189, 2012. ISSN 1573-2886. doi: 10.1007/s10878-011-9449-4. URL http://dx.doi.org/10.1007/s10878-011-9449-4.

[3] C. Bazgan, C. Bentz, C. Picouleau, and B. Ries. Blockers for the stability number and the chromatic number. *Graphs and Combinatorics*, 31(1):73–90, 2015. doi: 10.1007/s00373-013-1380-2. URL http://dx.doi.org/10.1007/s00373-013-1380-2.

[4] C. Bazgan, A. Nichterlein, and R. Niedermeier. A refined complexity analysis of finding the most vital edges for undirected shortest paths. In V. T. Paschos and P. Widmayer, editors, *Algorithms and Complexity - 9th International Conference, CIAC 2015, Paris, France, May 20-22, 2015. Proceedings*, volume 9079 of *Lecture Notes in Computer Science*, pages 47–60. Springer, 2015. ISBN 978-3-319-18172-1. doi: 10.1007/978-3-319-18173-8_3. URL http://dx.doi.org/10.1007/978-3-319-18173-8_3.

[5] H. L. Bodlaender, B. M. P. Jansen, and S. Kratsch. Kernelization lower bounds by cross-composition. *SIAM J. Discrete Math.*, 28(1):277–305, 2014.

[6] A. Brandstädt. Partitions of graphs into one or two independent sets and cliques. *Discrete Mathematics*, 152(1–3):47 – 54, 1996.

[7] A. Chateau and R. Giroudeau. Complexity and Polynomial-Time Approximation Algorithms around the Scaffolding Problem. In *Proc. AlCoB '14*, volume 8542 of *LNCS*, pages 47–58. Springer, 2014. ISBN 978-3-319-07952-3.

[8] A. Chateau and R. Giroudeau. A complexity and approximation framework for the maximization scaffolding problem. *Theoretical Computer Science*, 595:92 – 106, 2015.

[9] J. Chen, B. Chor, M. Fellows, X. Huang, D. W. Juedes, I. A. Kanj, and G. Xia. Tight lower bounds for certain parameterized NP-hard problems. *Inf. Comput.*, 201(2):216–231, 2005.

[10] P. Crescenzi. A short guide to approximation preserving reductions. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity, Ulm, Germany, June 24-27, 1997*, pages 262–273, 1997.

[11] A. Dayarian, T. Michael, and A. Sengupta. SOPRA: Scaffolding algorithm for paired reads via statistical optimization. *BMC Bioinformatics*, 11:345, 2010.

[12] R. G. Downey and M. R. Fellows. *Fundamentals of Parameterized Complexity*. Texts in Computer Science. Springer, 2013.

[13] S. Gao, W.-K. Sung, and N. Nagarajan. Opera: Reconstructing Optimal Genomic Scaffolds with High-Throughput Paired-End Sequences. *Journal of Computational Biology*, 18(11):1681–1691, 2011.

[14] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. publisher-freeman, 1979.

[15] J. Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(38), 1997.

[16] M. Hunt, C. Newbold, M. Berriman, and T. Otto. A comprehensive evaluation of assembly scaffolding tools. *Genome Biology*, 15(3):R42, 2014.

[17] R. Impagliazzo and R. Paturi. On the Complexity of k-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001.

[18] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? *J. Comput. Syst. Sci.*, 63 (4):512–530, 2001.

[19] D. Lokshtanov, D. Marx, and S. Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of the EATCS*, 105:41–72, 2011.

[20] J. Plesník. The np-completeness of the hamiltonian cycle problem in planar digraphs with degree bound two. *Inf. Process. Lett.*, 8(4):199–201, 1979. doi: 10.1016/0020-0190(79)90023-1. URL http://dx.doi.org/10.1016/0020-0190(79)90023-1.

[21] M. Weller, A. Chateau, and R. Giroudeau. On the complexity of scaffolding problems: From cliques to sparse graphs. In Z. Lu, D. Kim, W. Wu, W. Li, and D. Du, editors, *Combinatorial Optimization and Applications - 9th International Conference, COCOA 2015, Houston, TX, USA, December 18-20, 2015, Proceedings*, volume 9486 of *Lecture Notes in Computer Science*, pages 409–423. Springer, 2015. ISBN 978-3-319-26625-1. doi: 10.1007/978-3-319-26626-8_30. URL http://dx.doi.org/10.1007/978-3-319-26626-8_30.

[22] M. Weller, A. Chateau, and R. Giroudeau. Exact approaches for scaffolding. *BMC Bioinformatics*, 16(Suppl 14):S2, 2015. ISSN 1471-2105. doi: 10.1186/1471-2105-16-S14-S2. URL http://www.biomedcentral.com/1471-2105/16/S14/S2.

[23] M. Weller, A. Chateau, and R. Giroudeau. On the implementation of polynomial-time approximation algorithms for scaffold problems. *Technical Report*, 2015.

[24] G. Woeginger. Exact algorithms for np-hard problems: A survey. In *Combinatorial Optimization — Eureka, You Shrink!*, volume 2570 of *Lecture Notes in Computer Science*, pages 185–207. Springer Berlin Heidelberg, 2003.

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