# Solving the 2-Disjoint Connected Subgraphs Problem Faster Than $2^n$

Marek Cygan[1,*], Marcin Pilipczuk[1,**],
Michał Pilipczuk[2,***], and Jakub Onufry Wojtaszczyk[3,†]

[1] Institute of Informatics, University of Warsaw, Poland
{cygan,malcin}@mimuw.edu.pl
[2] Department of Informatics, University of Bergen, Norway
michal.pilipczuk@ii.uib.no
[3] Google Inc., Warsaw, Poland
onufry@google.com

**Abstract.** The 2-DISJOINT CONNECTED SUBGRAPHS problem, given a graph along with two disjoint sets of terminals $Z_1, Z_2$, asks whether it is possible to find disjoint sets $A_1, A_2$, such that $Z_1 \subseteq A_1$, $Z_2 \subseteq A_2$ and $A_1, A_2$ induce connected subgraphs. While the naive algorithm runs in $O(2^n n^{O(1)})$ time, solutions with complexity of form $O((2-\varepsilon)^n)$ have been found only for special graph classes [15,19]. In this paper we present an $O(1.933^n)$ algorithm for 2-DISJOINT CONNECTED SUBGRAPHS in general case, thus breaking the $2^n$ barrier. As a counterpoise of this result we show that if we parameterize the problem by the number of non-terminal vertices, it is hard both to speed up the brute-force approach and to find a polynomial kernel.

## 1 Introduction

It is commonly believed that no NP-complete problem is solvable in polynomial time. However, while all NP-complete problems are equivalent with respect to polynomial time reductions, they appear to be very different with respect to the best exponential time exact solutions. In particular, a number of NP-complete problems can be solved much faster than the obvious algorithm that enumerates all possible solutions; examples are INDEPENDENT SET [13], DOMINATING SET [13,21], CHROMATIC NUMBER [3] and BANDWIDTH [8]. The race for the fastest exact algorithm inspired several very interesting tools and techniques such as Fast Subset Convolution [2] and Measure&Conquer [13] (for an overview of the field we refer the reader to a recent book by Fomin and Kratsch [11]).

For several problems, including TSP, Chromatic Number, Permanent, Set Cover, #Hamiltonian Cycles and SAT, the currently best known time complexity is of the form $O(2^n n^{O(1)})$, which is a result of applying dynamic programming over subsets, the inclusion-exclusion principle or a brute force search. The question remains, however, which of those problems are inherently so hard that it is not possible to break the $2^n$ barrier and which are just waiting for new tools and techniques still to be discovered. In particular, the hardness of the $k$-SAT problem is the starting point for the Strong Exponential Time Hypothesis of Impagliazzo and Paturi [16,5], which is used as an argument that other problems are hard [7,17,18,6]. Recently, on the positive side, $O(c^n)$ time algorithms for a constant $c < 2$ have been developed for Sched [9], Maximum Induced Planar Subgraph [12] and (a major breakthrough in the field) for the undirected version of the Hamiltonian Cycle problem [1]. In most cases breaking the $2^n$-barrier gives an $O(c^n)$-time algorithm with $c$ significantly smaller than 2.

In this paper we study the 2-Disjoint Connected Subgraphs problem:

---

**2-Disjoint Connected Subgraphs**
**Input:** An undirected graph $G = (V, E)$ together with two disjoint subsets of vertices $Z_1, Z_2 \subseteq V$.
**Task:** Is it possible to find two disjoint subsets $A_1$ and $A_2$ of $V$, such that $Z_1 \subseteq A_1$, $Z_2 \subseteq A_2$ and $G[A_1]$, $G[A_2]$ are connected?

---

We call vertices from $Z_1 \cup Z_2$ *terminals* and all other vertices *non-terminals*. A general version of this problem (that is when we consider arbitrary number of sets) was used as one of tools in the celebrated result of Robertson and Seymour [20], that the Minor Containment problem can be solved in polynomial time for every fixed pattern graph H. In literature, the solution $(A_1, A_2)$ is sometimes required to be a partition of $V$. Note that this does not simplify the problem: in our setting, if $G$ is connected, the superfluous vertices can always be attached to one of the sets $A_1$ or $A_2$, while otherwise the instance is trivial.

*Previous work.* Gray et al. [14] show that the 2-Disjoint Connected Subgraphs problem is NP-complete even for the class of planar graphs. The motivation for this variant comes from an application in computational geometry, namely finding a realization of an imprecise terrain that minimizes the total number of local minima and local maxima. Furthermore, van't Hof et al. [15] proved that the problem remains NP-complete even if $|Z_1| = 2$.

From the exact exponential time algorithms perspective the authors in [15] show that one can solve 2-Disjoint Connected Subgraphs in $O((2 - \varepsilon(\ell))^n)$ time for any $n$-vertex $P_\ell$-free graph. This line of research was continued by Paulusma and van Rooij [19], where the authors present an algorithm with $O(1.2051^n)$ running time complexity for $P_6$-free graphs and split graphs.

*Our results.* In [19] Paulusma and van Rooij ask whether it is possible to solve the 2-Disjoint Connected Subgraphs problem in general graphs in faster than $O(2^n n^{O(1)})$ running time. In this paper we answer this question affirmatively and present an $O(1.933^n)$ time and polynomial space algorithm. Our approach

is based on the branch and reduce technique and a combinatorial study of the case when the number of terminal vertices is small.

**Theorem 1.** *One can solve the* 2-Disjoint Connected Subgraphs *problem in $O(1.933^n)$ running time and polynomial space.*

To break the $2^n$ barrier we heavily use the fact that in the instance size $n$ we count not only non-terminals but also terminal vertices. Observe that the naive brute-force algorithm, which for every partition of $V \setminus (Z_1 \cup Z_2)$ checks whether $A_1 \setminus Z_1, A_2 \setminus Z_2$ can be found within its sides, runs in $O(2^k n^{O(1)})$ time, where $k$ is the number of non-terminals. Therefore, natural question arises: is it possible to obtain also an $O((2-\varepsilon)^k n^{O(1)})$ algorithm? Using a recent hardness result (under the Strong Exponential Time Hypothesis; see Section 2 for an exact statement) of Cygan et al. [6] for the Set Splitting problem, we show that this is not the case: breaking the $2^k$ barrier is hard.

**Theorem 2.** *There does not exist an epsilon $\varepsilon > 0$ and an algorithm running in $O((2 - \varepsilon)^k |V|^{O(1)})$ time for the* 2-Disjoint Connected Subgraphs *problem unless the Strong Exponential Time Hypothesis is false.*

We should note that the Strong Exponential Time Hypothesis is a very strong assumption, and not widely believed, as it is the case of its weaker cousin, the Exponential Time Hypothesis (which is still much stronger than the statement $P \neq NP$). However, as noted in [17], SETH-based lower bounds still make sense: one should probably wait with trying to break a SETH-hard barrier until a faster SAT algorithm is discovered.

Since the naive $O(2^k n^{O(1)})$ time algorithm for 2-Disjoint Connected Subgraphs shows that the problem is fixed-parameter tractable, a usual next step is to investigate the kernelization possibilities for the problem (see Section 2 for respective definitions). Using the kernelization hardness result of Dom et al. [10] for the Colour Hitting Set problem parameterized by the size of the universe we show that 2-Disjoint Connected Subgraphs parameterized by the number of non-terminal vertices is unlikely to admit a polynomial kernel.

**Theorem 3.** *The* 2-Disjoint Connected Subgraphs *problem parameterized by the number of non-terminals does not admit a polynomial kernel unless $NP \subseteq coNP/poly$.*

As a byproduct we prove kernelization hardness result for Set Splitting parameterized by the size of the universe, which is of independent interest.

> Set Splitting
> **Input:** A set system $(\mathcal{F}, U)$, where $\mathcal{F} \subseteq 2^U$.
> **Task:** Does there exists a subset $X \subseteq U$ such that each set in $\mathcal{F}$ contains both an element from $X$ and from $U \setminus X$?

**Theorem 4.** *The* Set Splitting *problem parameterized by the universe size does not admit a polynomial kernel unless $NP \subseteq coNP/poly$.*

*Organization of the paper.* In Section 2 we provide the reader with the necessary definitions. Section 3 is devoted to the details of the $O(1.933^n)$ algorithm for 2-Disjoint Connected Subgraphs, while in Section 4 we present the negative results for the parameterization by the number of non-terminals. Concluding remarks are gathered in Section 5.

## 2   Preliminaries

*Notation.* For an undirected graph $G = (V, E)$ and a subset of vertices $X \subseteq V$ by $G[X]$ we denote the subgraph induced by $X$. An *articulation point* in $G$ is a vertex, whose removal increases the number of connected components of $G$.

For a universe $U$ and two subsets $X, Y \subseteq U$ we say that $X$ *splits* $Y$ if $Y \cap X \neq \emptyset$ and $Y \cap (U \setminus X) \neq \emptyset$.

*Parameterized complexity.* In the parameterized complexity setting, an instance comes with an integer parameter $k$ — formally, a parameterized problem $Q$ is a subset of $\Sigma^* \times \mathbb{N}$ for some finite alphabet $\Sigma$. We say that the problem is *fixed parameter tractable* (*FPT*) if there exists an algorithm solving any instance $(x, k)$ in time $f(k)\text{poly}(|x|)$ for some (usually exponential) computable function $f$. It is known that a problem is FPT iff it is kernelizable: a kernelization algorithm for a problem $Q$ takes an instance $(x, k)$ and in time polynomial in $|x| + k$ produces an equivalent instance $(x', k')$ (i.e., $(x, k) \in Q$ iff $(x', k') \in Q$) such that $|x'| + k' \leq g(k)$ for some computable function $g$. The function $g$ is the *size of the kernel*, and if it is polynomial, we say that $Q$ admits a polynomial kernel.

The hardness of kernelization can be obtained in a reduction-like manner by polynomial-time transformations from problems known to be hard to kernelize.

**Definition 5 ([4]).** *Let $P$ and $Q$ be parameterized problems. We say that $P$ is polynomial parameter reducible to $Q$, written $P \leq_p Q$, if there exists a polynomial time computable function $f : \Sigma^* \times \mathbb{N} \rightarrow \Sigma^* \times \mathbb{N}$ and a polynomial $p$, such that for all $(x, k) \in \Sigma^* \times \mathbb{N}$ the following holds: $(x, k) \in P$ iff $(x', k') = f(x, k) \in Q$ and $k' \leq p(k)$. The function $f$ is called a* polynomial parameter transformation.

**Theorem 6 ([4]).** *Let $P$ and $Q$ be parameterized problems and $\tilde{P}$ and $\tilde{Q}$ be the unparameterized versions of $P$ and $Q$ respectively. Suppose that $\tilde{P}$ is NP-hard and $\tilde{Q}$ is in NP. Assume there is a polynomial parameter transformation from $P$ to $Q$. Then if $Q$ admits a polynomial kernel, so does $P$.*

*Exponential Time Hypothesis.* [16,5] Let $c_k$ be the infimum of a set of reals $c$ for which there exists an algorithm solving the $k$-CNF-SAT problem in time $O(c^n)$, where $n$ is the number of variables in the input formula. The Exponential Time Hypothesis (ETH) asserts that $c_3 > 1$, whereas the Strong Exponential Time Hypothesis (SETH) asserts that $\lim_{k \to \infty} c_k = 2$. In particular, SETH implies that satisfiability of an arbitrary boolean formula cannot be checked significantly faster than by trying all possible assignments.

## 3   Algorithm

In this section we provide an algorithm solving the 2-Disjoint Connected Subgraphs problem in $O(1.933^n)$ time, where $n$ is the total number of vertices in the input graph. Recall that in the 2-Disjoint Connected Subgraphs problem the vertices were divided into three groups — $A_1$, $A_2$ and $V \setminus (A_1 \cup A_2)$. The algorithm will be simpler to present if we merge $A_2$ with $V \setminus (A_1 \cup A_2)$. This means that we drop the condition that $A_2$ is connected, requiring instead that all vertices from $Z_2$ are in the same connected component of $G[A_2]$. Formally, we consider the following problem:

---

**Modified 2-Disjoint Connected Subgraphs**
**Input:** An undirected graph $G = (V, E)$ together with two disjoint subsets of vertices $Z_1, Z_2 \subseteq V$.
**Task:** Is it possible to partition $V$ into sets $A_1, A_2$, with $Z_1 \subseteq A_1$ and $Z_2 \subseteq A_2$, such that $G[A_1]$ is connected and all the vertices of $Z_2$ lie in the same connected component of $G[A_2]$?

---

First notice that if we fix $A_1$, then checking the correctness of a partition is a polynomial-time operation.

Let $\varepsilon = 0.0493$. If $|Z_1| + |Z_2| \geq \varepsilon |V|$, we can simply iterate over all possible choices of $A_1 \setminus Z_1$ — this is a subset of $V \setminus (Z_1 \cup Z_2)$, so we have an $O(2^{|V|-|Z_1|-|Z_2|} n^{O(1)}) \subseteq O(2^{(1-\varepsilon)n} n^{O(1)}) \subseteq O(1.933^n)$ algorithm solving the problem. From now on we assume $|Z_1| + |Z_2| < \varepsilon n$. Additionally we assume that $|Z_1| \leq |Z_2|$, which means $|Z_1| < \varepsilon n/2$.

We will be looking for a set $A_1$ which is minimal with respect to inclusion. Let us fix one such set. Let $G' = G[A_1]$. Notice that any vertex $r \in A_1 \setminus Z_1$ has to be an articulation point in $G'$, otherwise we could move it from $A_1$ to $A_2$ and still have a valid solution. We will prove the following lemma:

**Lemma 7.** *Let $G'$ be any graph with at least 2 vertices. Let $k$ be the number of vertices of $G'$ that are not articulation points, and let $l$ be the number of articulation points of degree at least three. Then $3k - 6 \geq l$.*

The proof has been postponed to Section 3.2.

Let us call all the vertices of $Z_1$ and those vertices of $A_1$ with degree larger than 2 in $G'$ jointly *branching points*. As $|Z_1| \leq \varepsilon n/2$, we have at most $2\varepsilon n - 6$ branching points in $G'$.

Notice that we can assume that no vertex in $A_1 \setminus Z_1$ is of degree 0 or 1 in $G[A_1]$ (because, again, we could remove it to achieve a smaller solution) — thus all the vertices that are not branching points have to be of degree two in $G[A_1]$.

We now pursue a branch and reduce algorithm. The state of our algorithm is an instance $(G = (V, E), Z_1, Z_2)$ of the Modified 2-Disjoint Connected Subgraphs problem, plus a set $A_2 \supseteq Z_2$ of vertices that we have already chosen to be in $A_2$. The running time of our algorithm depends on two numbers:

- $N = |V \setminus A_2|$ — the number of vertices of the graph not yet chosen to $A_2$;
- $K$ — the upper limit on the number of branching points in the graph.

Formally, we describe a procedure $\mathtt{solve}(G, Z_1, Z_2, A_2, K)$ that either:

1. reports a valid solution $(A_1{}', A_2{}')$ to the MODIFIED 2-DISJOINT CONNECTED SUBGRAPHS problem, such that $A_2 \subseteq A_2{}'$, or
2. reports NO, correctly claiming that the MODIFIED 2-DISJOINT CONNECTED SUBGRAPHS instance $(G, Z_1, Z_2)$ has no solution $(A_1{}', A_2{}')$, such that $A_2 \subseteq A_2{}'$ and $A_1{}'$ has at most $K$ branching points.

Note that the outcomes above are not exclusive: if the input MODIFIED 2-DISJOINT CONNECTED SUBGRAPHS instance has a solution, but only with more than $K$ branching points, the algorithm can report any of the answers.

Let $\delta = (|Z_1|+|Z_2|)/|V|$, recall we assume $\delta \leq \varepsilon$. The starting value of $N$ is no larger than $(1 - \delta/2)|V|$. Lemma 7 ensures us that $\mathtt{solve}(G, Z_1, Z_2, Z_2, 2\delta|V| - 6)$ reports a solution if and only if the MODIFIED 2-DISJOINT CONNECTED SUBGRAPHS instance $(G, Z_1, Z_2)$ is a YES-instance.

Let $T(N, K)$ be the worst-case cost of executing our algorithm for given values of $N$ and $K$. We need to prove that $T((1 - \delta/2)n, 2\delta n - 6) \leq 1.933^n$ for any $0 \leq \delta \leq \varepsilon$.

We will also describe a similar procedure $\mathtt{solve}^*(G, Z_1, Z_2, A_2, K, w)$, where in addition to the input graph as in the original problem, we have a single vertex $w$, adjacent to some $t \in Z_1$, and we already decided that $w \in A_1$ (we did not decide whether $w$ is a branching point or not). Let $T^*(N, K)$ denote the worst-case complexity of executing our algorithm for given values of $N$ and $K$ in this case. We will simultaneously prove the following two inequalities:

$$T(N, K) \leq 1.5901^N 8.109^K,$$

$$T^*(N, K) \leq 0.3711 \cdot 1.5901^N 8.109^K.$$

It is easy to check this suffices to prove Theorem 1. As all the inequalities presented further are not tight within very small constants, the obtained bound on the running time is in fact slightly better; therefore, we can suppress all the polynomial factors using this error.

## 3.1   Reduction Rules

We provide a set of reduction rules. The algorithm at each point finds the lowest-number reduction rule applicable and applies it. Finding and applying a single reduction rule will trivially be a polynomial-time operation.

During the algorithm we will assign vertices both to $A_1$ and $A_2$. A feature of our algorithm is that a vertex $v$ can be assigned to $A_1$ only if there is some $w \in Z_1$ adjacent to it. When we assign $v$ to $A_1$, we will represent this by contracting the edge $vw$. Note that the resultant instance is fully equivalent to the original — if $(A_1, A_2)$ is a valid solution in the reduced graph, then $(A_1 \cup \{v\}, A_2)$ is a valid solution in the original graph, and conversely if $(A_1, A_2)$ is a valid solution in the original graph, and $v \in A_1$, then $(A_1 \setminus v, A_2)$ is a valid solution in the reduced graph.

If we find a solution in any of the branches, we return it as the witness of the positive answer for the problem. If we find no solution, we return NO.

**Reduction rules for** `solve`$(G, Z_1, Z_2, A_2, K)$

**Reduction rule 1.** *If there are two vertices from $Z_1$ in different components of $G[V \setminus A_2]$, return NO from this branch.*

**Reduction rule 2.** *If $|Z_1| = 1$, we check whether all the vertices of $Z_2$ lie in the same connected component of $G[V \setminus Z_1]$. If yes, we report a solution $(Z_1, V \setminus Z_1)$. If no, we return NO from this branch.*

**Reduction rule 3.** *If $K < |Z_1|$, we return NO from this branch.*

**Reduction rule 4.** *If there are two adjacent vertices $t, t' \in Z_1$, contract the edge $tt'$ and reduce $K$ by one.*

**Reduction rule 5.** *If there is a vertex $w$ adjacent to some $t \in Z_1$ (necessarily $w \notin Z_1$), we branch. We will either put it in $A_2$ in the solution, or choose it to be a part of $A_1$ and proceed to the second variant of the problem.*

To check the complexity assumptions in this case, we verify that the running time after applying the reduction rule is at most the claimed bound on $T(N, K)$:

$$T(N - 1, K) + T^*(N, K) \leq 1.5901^{N-1}8.109^K + 0.3711 \cdot 1.5901^N 8.109^K$$
$$\leq 1.5901^N 8.109^K.$$

The correctness of all the above rules is trivial.

**Reduction Rules for** `solve`$^*(G, Z_1, Z_2, A_2, K, w)$**.** Recall that now we have a vertex $w$ which we already know to be in $A_1$, and which is adjacent to some $t \in Z_1$.

**Reduction rule 6.** *If $w$ is adjacent to some vertex $t' \in Z_1$ other than $t$, we contract the edges $tw$ and $t'w$, and decrease $K$ by one.*

Notice that as we contracted two vertices from $Z_1$ into a single vertex in this case, we have at least one branching point less in the reduced graph than in the original. To check the complexity assumptions in this case, we verify that the running time after applying the reduction rule is at most the claimed bound on $T^*(N, K)$:

$$T(N - 2, K - 1) \leq 1.5901^{N-2}8.109^{K-1} \leq 0.3711 \cdot 1.5901^N 8.109^K.$$

**Reduction rule 7.** *If $w$ has no neighbours outside $Z_1 \cup A_2$, return NO from this branch.*

This can be justified as follows: for any solution $A_1$ containing $w$, the set $A_1 \setminus w$ is also a valid solution, and we are looking only for inclusion-minimal sets $A_1$.

**Reduction rule 8.** *If $w$ has only a single neighbour $w'$ outside $Z_1 \cup A_2$, contract the edge $tw$ and proceed treating $w'$ as the new $w$.*

The justification of this is that again, if $w'$ were not to be a part of the solution, we could also remove $w$ from the solution.

**Reduction rule 9.** *If $w$ has more than one neighbour outside $Z_1 \cup A_2$, we branch. We either decide $w$ is a branching point (in which case we contract $tw$, reduce the limit on the number of branching points by one, and return to the original problem), or that it is not. In the latter case, exactly one of the neighbours of $w$ is in the solution — thus we branch on which one is it, contract the edge $tw$, put the remaining neighbours of $w$ in $A_2$ and proceed treating the selected neighbour as the new $w$.*

To check the complexity assumptions we verify that for any $d \geq 3$ the running time after applying the reduction rule is at most the claimed bound on $T^*(N, K)$:

$$T(N-1, K-1) + (d-1)T^*(N-(d-1), K)$$
$$\leq 1.5901^{N-1}8.109^{K-1} + (d-1)0.3711 \cdot 1.5901^{N-(d-1)}8.109^K$$
$$\leq 0.3711 \cdot 1.5901^N 8.109^K.$$

As $d/(d-1) \leq 1.5 < 1.5901$ for $d \geq 3$, the above inequality may be verified only for $d = 3$.

### 3.2    Proof of Lemma 7

**Lemma 8 (Lemma 7, restated).** *Let $G'$ be any graph with at least $2$ vertices. Let $k$ be the number of vertices of $G'$ that are not articulation points, and let $l$ be the number of articulation points of degree at least three. Then $3k - 6 \geq l$.*

*Proof.* Let us denote $m = 3k - l$, our goal is to prove that for any graph $G$ we have $m \geq 6$. Let us assume the contrary: there exist graphs for which $m < 6$. Let $G_0$ be the one with minimum number of vertices, a minimal counterexample. We now examine the structure of $G_0$ in order to find a contradiction.

We first claim that $G_0$ is connected. If $G_0$ is edgeless, $k \geq 2$, $l = 0$ and $3k - l \geq 6$. If $G_0$ contains an isolated vertex $v$, $G_0$ has at least three vertices, and by deleting $v$ we obtain a smaller counterexample, as we decrease $k$ and keep the value of $l$. Otherwise, we note that $m$ is additive with respect to connected components, so in case of disconnectedness one of its components would be a smaller counterexample with at least two vertices.

Observe that $G_0$ cannot have an articulation point of degree 2. In such a situation we could contract one of the incident edges; this operation does not change neither $k$ nor $l$, so $m$ stays the same and we obtain a smaller counterexample.

Now assume that $G_0$ contains an articulation point $v$, such that we can partition the connected components of $G_0[V \setminus \{v\}]$ into two parts with vertex sets $V_1, V_2$, for which $|V_1|, |V_2| > 1$. Note that $G_0[V_1 \cup \{v\}]$ and $G_0[V_2 \cup \{v\}]$ are connected. As the degree of $v$ is at least 3, let us assume without losing generality that the degree of $v$ in $G_0[V_1 \cup \{v\}]$ is at least 2. Consider two graphs:

- $G_1$, being $G_0[V_1 \cup \{v\}]$ with a pendant (a degree-1 vertex) attached to $v$;
- $G_2 := G_0[V_2 \cup \{v\}]$.

Both $G_1$ and $G_2$ have less vertices than $G_0$, so if for $i = 1, 2$ we denote by $k_i, l_i$ the numbers of non-articulation points and articulation points of degree at least 3 in $G_i$ respectively, then $m_i := 3k_i - l_i \geq 6$.

Observe that $k \geq k_1 + k_2 - 2$, as every vertex from $V \setminus \{v\}$ that was a non-articulation point in a corresponding $G_i$ is also a non-articulation point in $G_0$, while $v$ is an articulation point in $G_1$. Moreover, $l \leq l_1 + l_2$, as every articulation point of degree at least 3 in $G_0$ is also an articulation point of degree at least 3 in either $G_1$ or $G_2$. Hence,

$$m = 3k - l \geq 3(k_1 + k_2 - 2) - (l_1 + l_2) = m_1 + m_2 - 6 \geq 6,$$

a contradiction with $G_0$ being a counterexample.

Take any articulation point $v$ in $G_0$. As the partition described above is not possible, removing $v$ from the graph results in $G_0$ being split into exactly two connected components: one of size $|V(G)| - 2$ and one of size 1. Therefore, every articulation point in $G_0$ has a neighbour of degree 1, which obviously is not an articulation point. Hence, $l \leq k$ and, consequently, $6 > 3k - l \geq 2k$, which leads to $l \leq k \leq 2$. Now observe that

- if $l = 0$, then $k \geq 2$ as $G_0$ has at least two vertices, so $m \geq 6$, a contradiction;
- if $l = 1$, then the only articulation point must have at least three neighbours, so $k \geq 3$, a contradiction;
- otherwise $l = 2$ and the graph is isomorphic to $P_4$, a contradiction as well.

In each case we obtained a contradiction, which finishes the proof.     □

*Remark 9.* One might wonder whether the constants obtained in the proof are optimal, and whether we could strengthen the bound to achieve a better complexity of the algorithm. In fact, this is not the case: there exists an infinite family of graphs with unbounded $k, l$, for which the inequality from Lemma 7 is tight. An example of such a graph is one constructed from a set of disjoint triangles by connecting them in a tree-like manner by bridges, and by adding a degree-1 neighbours to some of the vertices of the triangles, such that each vertex of the triangles has degree exactly three.

## 4   2-Disjoint Connected Subgraphs Parameterized by the Number of Non-terminals

In this section we consider the 2-Disjoint Connected Subgraphs problem parameterized by the number of non-terminals and prove Theorems 2 and 3.

First we prove that the Set Splitting problem parameterized by the size of the universe does not admit a polynomial kernel unless NP $\subseteq$ coNP/poly(that is, prove Theorem 4), which we later use for the kernelization hardness of 2-Disjoint Connected Subgraphs. As the starting point we use the following result of Dom et al. [10].

Colour Hitting Set
**Input:** A set system $(\mathcal{F}, U)$, where $\mathcal{F} \subseteq 2^U$, an integer $k \leq |U|$ and a colour function $c : U \to \{0, \ldots, k-1\}$.
**Task:** Does there exists a subset $X \subseteq U$, which contains at most one element of each of the $k$ colours, such that each set in $\mathcal{F}$ contains an element from $X$?

**Theorem 10 ([10]).** *The* Colour Hitting Set *problem parameterized by the size of the universe does not have a polynomial kernel unless NP $\subseteq$ coNP/poly.*

**Lemma 11.** *There exists a polynomial time algorithm which given an instance* $I = (\mathcal{F}, U)$ *of* Colour Hitting Set *creates an instance* $I' = (\mathcal{F}', U')$ *of* Set Splitting*, where* $|U'| = |U| + 2$*, such that* $I$ *is a YES-instance iff* $I'$ *is a YES-instance.*

*Proof.* Let $I = (\mathcal{F}, U, k, c)$ be an instance of Colour Hitting Set. We create the following instance $I' = (\mathcal{F}', U')$ of Set Splitting. As the universe $U'$ we take $U' = U \cup \{u_{\mathrm{in}}, u_{\mathrm{out}}\}$. As the family $\mathcal{F}'$ we set $\mathcal{F}' = \{\{u_{\mathrm{out}}\} \cup S \; : \; S \in \mathcal{F}\} \cup \{\{u_{\mathrm{in}}, u_{\mathrm{out}}\}\} \cup \{\{u_{\mathrm{in}}, u_1, u_2\} \; : \; u_1, u_2 \in U \wedge u_1 \neq u_2 \wedge c(u_1) = c(u_2)\}$. We want to prove that $I$ is a YES-instance of Colour Hitting Set iff $I'$ is a YES-instance of Set Splitting.

First assume that $I$ is a YES-instance of Colour Hitting Set and let $X \subseteq U$ be any solution. We claim that $X' = X \cup \{u_{\mathrm{in}}\}$ is a solution to Set Splitting. Obviously we split the set $\{u_{\mathrm{in}}, u_{\mathrm{out}}\}$. Since each set in $\mathcal{F}$ contains an element of $X$ and $u_{\mathrm{out}} \notin X'$ we infer that each set of $\{\{u_{\mathrm{out}}\} \cup S \; : \; S \in \mathcal{F}\}$ is split by $X'$. Finally, $X$ contains at most one element of each colour, therefore each set $\{u_{\mathrm{in}}, u_1, u_2\}$ is split, because $u_{\mathrm{in}} \in X'$ and at least one element of $\{u_1, u_2\}$ belongs to $U' \setminus X'$.

Now let us assume that $I'$ is a YES-instance and let $X' \subseteq U'$ be any solution. Since by symmetry $U' \setminus X'$ is also a solution for $I'$, and there exists a set $\{u_{\mathrm{in}}, u_{\mathrm{out}}\}$ in $I'$, w.l.o.g. we assume that $u_{\mathrm{in}} \in X'$ and $u_{\mathrm{out}} \notin X'$. Let $X = X' \setminus \{u_{\mathrm{in}}\} \subseteq U$. We prove that $X$ is a solution for the instance $I$. Since each set in $\{\{u_{\mathrm{out}}\} \cup S \; : \; S \in \mathcal{F}\}$ is split by $X'$ and $u_{\mathrm{out}} \notin X'$ we infer that $X$ contains at least one vertex from each set in $\mathcal{F}$. Moreover, since each set of the form $\{u_{\mathrm{in}}, u_1, u_2\}$ is split and $u_{\mathrm{in}} \in X'$, we know that $X$ contains at most one element of each colour and, consequently, $I$ is a YES-instance. $\qed$

By Theorems 10, 6 and the above lemma Theorem 4 follows. Now we show a reduction from Set Splitting to 2-Disjoint Connected Subgraphs.

**Lemma 12.** *There exists a polynomial time algorithm which given an instance* $(\mathcal{F}, U)$ *of* Set Splitting *creates an instance* $(G = (V, E), Z_1, Z_2)$ *of* 2-Disjoint Connected Subgraphs*, where* $|V| - |Z_1| - |Z_2| = |U|$*, such that* $(\mathcal{F}, U)$ *is a YES-instance iff* $(G, Z_1, Z_2)$ *is a YES-instance.*

*Proof.* Let $(\mathcal{F}, U)$ be an instance of the Set Splitting problem. We construct the following graph $G = (V, E)$, where $V = \{z_1, z_2\} \cup U \cup \{v_1^S, v_2^S \; : \; S \in \mathcal{F}\}$, that

is as the vertex set we take two special vertices $z_1, z_2$, one vertex for each element of the universe $U$ and two vertices for each set of the family $\mathcal{F}$. We define the set of edges of the graph $G$ as $E = \{z_1 u, z_2 u \ : \ u \in U\} \cup \{v_1^S u, v_2^S u \ : \ S \in \mathcal{F} \wedge u \in S\}$. Finally we define $Z_1 = \{z_1\} \cup \{v_1^S \ : \ S \in \mathcal{F}\}$ and $Z_2 = \{z_2\} \cup \{v_2^S \ : \ S \in \mathcal{F}\}$, hence the only non-terminal vertices in $G$ are vertices of $U$.

Now we prove that $(\mathcal{F}, U)$ is a YES-instance of SET SPLITTING iff $(G = (V, E), Z_1, Z_2)$ is a YES-instance of 2-DISJOINT CONNECTED SUBGRAPHS. First assume that $X \subseteq U$ is a solution for $(\mathcal{F}, U)$. We set $A_1 = Z_1 \cup X$ and $A_2 = Z_2 \cup (U \setminus X)$. Since each set in $S \in \mathcal{F}$ contains both an element of $X$ and of $U \setminus X$, all the vertices from $A_1$ are connected to $z_1$ and all the vertices from $A_2$ are connected to $z_2$. Therefore, the subgraphs $G[A_1]$ and $G[A_2]$ are connected and, consequently, $(G, Z_1, Z_2)$ is a YES-instance.

In the other direction assume that $(G = (V, E), Z_1, Z_2)$ is a YES-instance and let $(A_1, A_2)$ be a solution. We claim that $X = A_1 \cap U$ is a solution to $(\mathcal{F}, U)$. Indeed, since both $G[A_1]$ and $G[A_2]$ are connected, we infer that each set $S \in \mathcal{F}$ contains both an element of $X$ and of $U \setminus X$; otherwise either vertices $z_1, v_1^S$ or $z_2, v_2^S$ would be disconnected in $G[A_1]$ and $G[A_2]$ respectively. $\qquad\square$

To prove Theorem 2 we pipeline Lemma 12 with the following recent result:

**Theorem 13 ([6]).** *There does not exist an epsilon $\varepsilon > 0$ and an algorithm running in $(2 - \varepsilon)^{|U|}(|U| + |\mathcal{F}|)^{O(1)}$ time for the* SET SPLITTING *problem unless the Strong Exponential Time Hypothesis is false.*

*Remark 14.* The proof of Theorem 13 in [6] reduces a $\kappa$-CNF-SAT instance (for any $\kappa$) to a SET SPLITTING instance with a number of sets linear in the number of clauses in the input. If we pipeline this reduction with Lemma 12, we obtain an instance with much more terminals than non-terminals: $k$ non-terminals yields roughly $k^\kappa$ terminals. Therefore there is much space for improvement between the instances produced by this reduction and the ones where our $O(1.933^n)$-time algorithm does not fall back to the brute-force solution.

## 5 Conclusions

In this paper we studied the general case of 2-DISJOINT CONNECTED SUB-GRAPHS problem and showed an algorithm achieving running time $O(1.933^n)$, thus breaking the $2^n$ barrier. For the natural parameterization by the number of non-terminals we have shown a matching lower bound for the naive $O(2^k n^{O(1)})$ upper bound. Moreover, the existence of a polynomial kernel is unlikely.

The obvious question is whether we can develop a significantly faster algorithm than $O(1.933^n)$. It is possible that a deeper combinatorial study and enhancement of our branching rules can lead to such a result.

## References

1. Björklund, A.: Determinant sums for undirected hamiltonicity. In: 51th Annual IEEE Symposium on Foundations of Computer Science (FOCS), pp. 173–182 (2010)

2. Björklund, A., Husfeldt, T., Kaski, P., Koivisto, M.: Fourier meets möbius: fast subset convolution. In: 39th Annual ACM Symposium on Theory of Computing (STOC), pp. 67–74 (2007)
3. Björklund, A., Husfeldt, T., Koivisto, M.: Set partitioning via inclusion-exclusion. SIAM J. Comput. 39(2), 546–563 (2009)
4. Bodlaender, H.L., Thomasse, S., Yeo, A.: Analysis of data reduction: Transformations give evidence for non-existence of polynomial kernels, technical Report UU-CS-2008-030, Institute of Information and Computing Sciences, Utrecht University, Netherlands (2008)
5. Calabro, C., Impagliazzo, R., Paturi, R.: The Complexity of Satisfiability of Small Depth Circuits. In: Chen, J., Fomin, F.V. (eds.) IWPEC 2009. LNCS, vol. 5917, pp. 75–85. Springer, Heidelberg (2009)
6. Cygan, M., Dell, H., Lokshtanov, D., Marx, D., Nederlof, J., Okamoto, Y., Paturi, R., Saurabh, S., Wahlström, M.: On problems as hard as CNF-SAT. CoRR abs/1112.2275 (2011)
7. Cygan, M., Nederlof, J., Pilipczuk, M., Pilipczuk, M., van Rooij, J.M.M., Wojtaszczyk, J.O.: Solving connectivity problems parameterized by treewidth in single exponential time. In: 52th Annual IEEE Symposium on Foundations of Computer Science, FOCS (2011) (to appear)
8. Cygan, M., Pilipczuk, M.: Exact and approximate bandwidth. Theor. Comput. Sci. 411(40-42), 3701–3713 (2010)
9. Cygan, M., Pilipczuk, M., Pilipczuk, M., Wojtaszczyk, J.O.: Scheduling Partially Ordered Jobs Faster Than $2^n$. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 299–310. Springer, Heidelberg (2011)
10. Dom, M., Lokshtanov, D., Saurabh, S.: Incompressibility through Colors and IDs. In: Albers, S., Marchetti-Spaccamela, A., Matias, Y., Nikoletseas, S., Thomas, W. (eds.) ICALP 2009, Part I. LNCS, vol. 5555, pp. 378–389. Springer, Heidelberg (2009)
11. Fomin, F.V., Kratsch, D.: Exact Exponential Algorithms. Springer (2010)
12. Fomin, F.V., Todinca, I., Villanger, Y.: Exact Algorithm for the Maximum Induced Planar Subgraph Problem. In: Demetrescu, C., Halldórsson, M.M. (eds.) ESA 2011. LNCS, vol. 6942, pp. 287–298. Springer, Heidelberg (2011)
13. Fomin, F.V., Grandoni, F., Kratsch, D.: A measure & conquer approach for the analysis of exact algorithms. J. ACM 56(5), 1–32 (2009)
14. Gray, C., Kammer, F., Löffler, M., Silveira, R.I.: Removing Local Extrema from Imprecise Terrains. CoRR abs/1002.2580 (2010)
15. van 't Hof, P., Paulusma, D., Woeginger, G.J.: Partitioning graphs into connected parts. Theor. Comput. Sci. 410(47-49), 4834–4843 (2009)
16. Impagliazzo, R., Paturi, R.: On the complexity of k-SAT. J. Comput. Syst. Sci. 62(2), 367–375 (2001)
17. Lokshtanov, D., Marx, D., Saurabh, S.: Known Algorithms on Graphs of Bounded Treewidth are Probably Optimal. In: Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 777–789 (2011)
18. Patrascu, M., Williams, R.: On the possibility of faster SAT algorithms. In: Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pp. 1065–1075 (2010)
19. Paulusma, D., van Rooij, J.M.M.: On partitioning a graph into two connected subgraphs. Theor. Comput. Sci. 412(48), 6761–6769 (2011)
20. Robertson, N., Seymour, P.D.: Graph minors XIII. The disjoint paths problem. J. Comb. Theory, Ser. B 63(1), 65–110 (1995)
21. van Rooij, J.M.M., Nederlof, J., van Dijk, T.C.: Inclusion/Exclusion Meets Measure and Conquer. In: Fiat, A., Sanders, P. (eds.) ESA 2009. LNCS, vol. 5757, pp. 554–565. Springer, Heidelberg (2009)