# Connected Search for a Lazy Robber

**Isolde Adler**
School of Computing, University of Leeds, Leeds, United Kingdom
I.M.Adler@leeds.ac.uk

**Christophe Paul**
CNRS, LIRMM, Université de Montpellier, Montpellier, France
christophe.paul@lirmm.fr

**Dimitrios M. Thilikos**
CNRS, LIRMM, Université de Montpellier, Montpellier, France
sedthilk@thilikos.info

## Abstract

The node search game against a lazy (or, respectively, agile) invisible robber has been introduced as a search-game analogue of the treewidth parameter (and, respectively, pathwidth). In the *connected* variants of the above two games, we additionally demand that, at each moment of the search, the *clean* territories are *connected*. The connected search game against an agile and invisible robber has been extensively examined. The monotone variant (where we also demand that the clean territories are progressively increasing) of this game, corresponds to the graph parameter of *connected pathwidth*. It is known that *the price of connectivty* to search for an agile robber is bounded by 2, that is the connected pathwidth of a graph is at most twice (plus some constant) its pathwidth. In this paper, we investigate the connected search game against a *lazy* robber. A lazy robber moves only when the searchers' strategy threatens the location that he currently occupies. We introduce two alternative graph-theoretic formulations of this game, one in terms of connected tree decompositions, and one in terms of (connected) layouts, leading to the graph parameter of *connected treewidth*. We observe that connected treewidth parameter is closed under contractions and prove that for every $k \geq 2$, the set of contraction obstructions of the class of graphs with connected treewidth at most $k$ is infinite. Our main result is a complete characterization of the obstruction set for $k = 2$. One may observe that, so far, only a few complete obstruction sets are explicitly known for contraction closed graph classes. We finally show that, in contrast to the agile robber game, the price of connectivity is unbounded.

**2012 ACM Subject Classification** Mathematics of computing → Graph theory

**Keywords and phrases** Graph searching, Tree-decomposition, Obstructions

**Digital Object Identifier** 10.4230/LIPIcs.FSTTCS.2019.7

**Funding** *Christophe Paul*: Supported by projects DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010).
*Dimitrios M. Thilikos*: Supported by projects DEMOGRAPH (ANR-16-CE40-0028) and ESIGMA (ANR-17-CE23-0010). Supported by the Research Council of Norway and the French Ministry of Europe and Foreign Affairs, via the Franco-Norwegian project PHC AURORA 2019.

## 1 Introduction

A *graph-search game* is opposing a group of searchers and a robber that are moving in turn on a graph. A *search strategy* is a sequence of moves of the searchers that eventually leads to the capture of the robber. The *cost* of a search strategy is the maximum number of searchers simultaneously present on the graph during the search strategy. The *search number* of a graph is defined as the minimum cost of a search strategy. Different rules imposed on the search strategy and the moves of the robber define different searching games. The


39th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science (FSTTCS 2019).
Editors: Arkadev Chattopadhyay and Paul Gastin; Article No. 7; pp. 7:1–7:14
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

study of graph searching parameters is an active field of graph theory as several important graph parameters have their search-game analogues that provide useful insights. For related surveys, see [2, 3, 10, 21, 38].

One of the most classic graph-search games is the one of *node-search* introduced by Kirousis and Papadimitriou [31, 32]. In this version, both the searchers and the robber occupy vertices of the graph. One searcher can move at a time. The capture of the robber happens when some searcher and the robber simultaneously occupy the same vertex and that the robber cannot escape along a path free of a searcher. In this paper we consider *monotone* search strategies against an *invisible* robber. Being invisible implies that the search strategy has to be independent of the moves of the robber. A search strategy is monotone if it prevents the robber from moving to vertices that have been already occupied by the searchers, implying that the *robber territory* is never increasing. The robber territory is the set of vertices that can be reached from the robber position by a path free of searcher.

**Agility and laziness.** A robber can be *lazy* or *agile*. A lazy robber resides on a vertex as long as a searcher is not placed on that vertex, while an agile robber may move whenever he wants to. The distinction between a lazy and an agile robber was introduced for the first time in [13]. Motivated by established links with well-studied graph theoretical parameters, there is an extensive amount of research on the different variants of the search game depending on the monotonicity constraint and on the laziness or agility of the robber. In particular, the monotone search number of a graph $G$ against an agile (resp. lazy) robber is equal to the pathwidth (resp. treewidth) of $G$ [13, 31, 32, 36, 42]. Also, it was proven that the non-monotone variants are equal to their monotone counterparts [8, 9, 20, 34, 42].

**The connectivity issue.** In both search games described above, no constraint (apart from the monotonicity, which in this context, as mentioned before, is no restriction) is ruling the move of a searcher. That is, a searcher can move arbitrarily far away from his/her original position. For this reason, such search games have been called “helicopter search games” (as suggested in [42]). From the application view point, this teleportation ability is not always realistic. In some settings (like cave exploration), it is natural to constrain the search to be connected. That is, the clean territory induces a connected subgraph[1] at each step of the search (see [24] for an example).

This inspired the question on the “price of connectivity”, asking whether there is some universal constant $c$ such that the connected search number is no more than $c$ times its non-connected counterpart. In its original form, this question was asked in [5] for the agile variant and, in the same paper, it was answered affirmatively for the case of trees (see also [6,16–19,22,37] for related results). Later, it was proved for all graphs by Dereniowski [14], who suggested a connected counterpart of pathwidth, called connected pathwidth, that is equivalent to the monotone connected agile search number. Then it was proved that this parameter is always upper bounded by twice the pathwidth plus one.

[1] Interestingly, the motivating story of one of the foundational articles on graph searching, authored by Torrence Parsons [39] in 1976, was inspired by an earlier article of Breisch in *Southwestern Cavers Journal* [11] proposing a “speleotopological” approach for the problem of finding an explorer lost in a system of dark caves. It is worth to stress that this setting neglected the natural connectivity requirement.

## 1.1 Our contributions

**Connected treewidth.** In this paper, we study the (monotone) connected search against a *lazy* robber. Our first contribution is to establish the parameter by giving two alternative definitions: one in terms of *connected* tree decompositions and one in terms of *connected* layouts. Intuitively, a tree-decomposition $(T, \mathcal{F})$ is connected[2] if it can be rooted at some node $r$ in a way that for every node $u$, the subgraph $G_u$, induced by the subset $V_u$ of vertices appearing in some bag on the path in $T$ between $r$ and $u$, is connected. We observe that this is a natural extension of the concept of connected pathwidth proposed by [14]. Our layout definition is a variant of the classic layout definition of [13] with the restriction that now we only consider layouts where every prefix induces a connected graph. Our equivalence, proven in Section 3, indicates that monotone connected search against a *lazy* robber can be seen as a natural way to define a *connected* version of treewidth. We also stress that the non-monotone variant of this game corresponds to an different parameter, as proved in [24]. Yet another way to define "connected" treewidth is to consider tree decompositions where for every $t \in V_T$, the bag $X_t$ induces a connected subgraph of $G$. We refer to this variant *bag-connected treewidth* (while the one we define in this paper can be called *prefix-connected treewidth*). Bag-connected treewidth was introduced independently by Jégou and Terrioux in [27], in the context of solving Constraint Satisfaction Problems (CSPs)(see [26, 28]) and, in a combinatorial context, by Diestel and Müller in [15] who revealed interesting relations with graph-geometric parameters such as the geodesic cycle number, graph hyperbolicity (see also [25]).

**Contraction Obstructions.** We say that a graph $H$ is a *contraction* of $G$, denoted by $H \preceq G$, if a graph isomorphic to $H$ can be obtained from $G$ by a series of *edge contractions*. We also say that $H$ is a *minor* of $G$ if $H$ is a contraction of a subgraph of $G$. We define the *minor obstructions* (*contraction obstructions*, respectively) of a graph class $\mathcal{G}$, denoted by $\mathbf{obs}_{\leq}(\mathcal{G})$ ($\mathbf{obs}_{\preceq}(\mathcal{G})$, respectively), as the set of all minor (contraction, respectively) minimal graphs that do not belong to $\mathcal{G}$. It is easy to see that when $\mathcal{G}$ is minor (contraction, respectively) closed, then $\mathbf{obs}_{\leq}(\mathcal{G})$ ($\mathbf{obs}_{\preceq}(\mathcal{G})$, respectively) provides a *complete* characterization of a minor closed (contraction, respectivelly) class $\mathcal{G}$: a graph belongs to $\mathcal{G}$ if and only if it excludes all graphs in $\mathbf{obs}_{\leq}(\mathcal{G})$ (respectively $\mathbf{obs}_{\preceq}(\mathcal{G})$) as minors (contractions, respectively). Moreover, in the case of the minor relation, we know from the theorem of Roberston and Seymour [40] that the set $\mathbf{obs}_{\leq}(\mathcal{G})$ is always finite and therefore the aforementioned characterization provides a *finite characterization* of any minor closed class in terms or forbidden minors. To identify (or even to compute) $\mathbf{obs}_{\leq}(\mathcal{G})$ for different instantiations of minor closed graph classes is an interesting topic in graph theory (see [1, 35]). For instance, if $\mathcal{T}_k$ is the class of graphs with treewidth at most $k$, then $\mathbf{obs}_{\leq}(\mathcal{T}_k)$ is known for every $k \leq 3$ [4] and remains unknown for $k > 3$ (see [41] for some partial results for the case where $k = 4$). Similarly, if $\mathcal{P}_k$ is the class of graphs with pathwidth at most $k$, then $\mathbf{obs}_{\leq}(\mathcal{P}_k)$ is known for $k \leq 2$ [30] and remains unknown for $k > 2$. Bounds for the size of the graphs in $\mathbf{obs}_{\leq}(\mathcal{T}_k)$ and $\mathbf{obs}_{\leq}(\mathcal{P}_k)$ have been proved in [33].

Unfortunately, the landscape is more obscure for the contraction relation as contraction obstruction sets are not finite in general. Contraction obstruction sets are only known for a few contraction closed classes. For instance, the contraction obstruction set for planar

[2] We also want to point out that alternative notions of connected tree-decomposition have been considered, see for example [23] and [15, 27] for two different definitions. We believe that the parameter correspondence we establish is a strong argument in favour of our definition proposal.

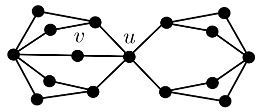


**Figure 1** A graph $G \in \mathcal{T}_2^c$ such $G - uv \notin \mathcal{T}_2^c$ and $G - v \notin \mathcal{T}_2^c$.

graphs is described in [12]. A more elaborate example of a finite contraction obstruction set was identified in [7], containing 177 connected graphs, for the class of graphs whose connected mixed search number (for an agile and invisible robber) is at most 2. Another class characterized by an infinite set of contraction obstructions is discussed in [29].

Let $k \in \mathbb{N}$. By $\mathcal{T}_k^c$, we denote the class of all (connected) graphs with connected treewidth at most $k$. We observe that $\mathcal{T}_2^c$ is not minor closed: removing a vertex or an edge (see e.g., the graph $G$ of Figure 1) may increase the connected treewidth. Therefore, no characterization via minor obstruction exists. However, in this paper we observe that $\mathcal{T}_k^c$ is contraction closed, for every $k$, and it is a challenging problem to identify $\mathcal{O}_k := \mathbf{obs}_{\preceq}(\mathcal{T}_k^c)$ for distinct values of $k$, especially since we have no guarantee that this set is finite. Moreover, in case $\mathcal{O}_k$ is infinite, we are essentially looking for a finite canonical description of this set.

Our second contribution is the complete identification of $\mathcal{O}_2$. As a preliminary part of our results, in Subsection 4.1, we prove general properties of $\mathcal{O}_k$ for every $k$. These are later used to identify $\mathcal{O}_2$. In Section 5, that is the most technical part of this paper, we prove that $\mathcal{O}_2$ is an infinite set that can be canonically described by a sequence of gluing operations.

**Price of connectivity.** We give, for every $k \geq 2$, an *infinite* subset of $\mathbf{obs}_{\preceq}(\mathcal{T}_k^c)$ consisting of graphs of treewidth 2, i.e., graphs in $\mathcal{T}_2$ (see Section 4.2). Consequently, the price of connectivity on treewidth is unbounded and this makes a sharp contrast with the corresponding result on pathwidth. To conclude, for monotone search, the price of connectivity is bounded when we are searching for an agile robber while this price goes to infinity when the robber is lazy. This latter contribution provides a simpler construction of a result from [24] that the cost of connectivity can be log n, where n is the number of vertices.

# 2 Preliminaries

## 2.1 Standard definitions

**Sequences.** Given a finite set $U$, a *sequence* $\sigma$ over $U$ is a bijection $\sigma : U \to [|U|]$. For $x \in U$, $\sigma(x) = i$ if $x$ is at the $i$-th position in $\sigma$ and we denote $\sigma_i = \sigma^{-1}(i)$. For $x, y \in U$, if $\sigma(x) < \sigma(y)$, we write $x <_\sigma y$. We define the sets $\sigma_{<i} = \{x \in U \mid \sigma(x) < i\}$ and $\sigma_{\leq i} = \{x \in U \mid \sigma(x) \leq i\}$. Alternatively, we denote a sequence by $\sigma = \langle \sigma_1, \ldots, \sigma_n \rangle$.

**Graphs.** The graphs we consider are undirected are simple. We use standard notations. For a subset $S$ of vertices, $G[S]$ denotes the subgraph induced by $S$. A *separator* is a subset $S$ of vertices such that $G \setminus S = G[V \setminus S]$ contains more connected components than $G$. A connected component $H$ of $G \setminus S$ is a *full $S$-component* of $G$ if $N_G(V(H)) = S$. We denote by $\mathcal{C}(G, S)$ the set of all full $S$-connected components of $G$ and by $\mathcal{F}(G, S)$ the set containing every induced subgraph $G[S \cup C]$ with $C \in \mathcal{C}(G, S)$. The set of cut vertices of a graph $G$ is denoted $C(G)$.

*Contracting* an edge $e = xy \in E(G)$ yields the graph $G/e$ obtained by removing $x$ and $y$ from $G$, introducing a new vertex and making it adjacent with all vertices in $N_G(\{x,y\})\setminus\{x,y\}$. If $F$ is a subset of edges of $G$, then $G/F$ is the graph obtained by contracting the edges of $F$. We say that a graph $H$ is a *contraction* of $G$, denoted by $H \preceq G$, if a graph isomorphic to $H$ can be obtained by a series of edge contractions.

A *tree-decomposition* of a graph $G = (V,E)$ is a pair $(T,\mathcal{F})$ where $T = (V_T, E_T)$ is a tree and $\mathcal{F} = \{X_t \subseteq V \mid t \in V_T\}$ such that : 1) $\bigcup_{t\in V_T} X_t = V$; 2) for every edge $e \in E$, there exists a node $t \in T$ such that $e \subseteq X_t$; and 3) for every vertex $x \in V$, the set $\{t \in V_T \mid x \in X_t\}$ induces a connected subgraph of $T$. We refer to $V_T$ as the set of *nodes of* $T$ and the sets of $\mathcal{F}$ as the *bags* of $(T,\mathcal{F})$. The *width* of a tree-decomposition $(T,\mathcal{F})$ is $\mathsf{width}(T,\mathcal{F}) = \max\big\{|X|-1 \mid X \in \mathcal{F}\big\}$ and the *tree-width* of a graph $G$ is $\mathsf{tw}(G) = \min\big\{\mathsf{width}(T,\mathcal{F}) \mid (T,\mathcal{F})$ is a tree-decomposition of $G\big\}$.

**Rooted graphs.** A *$q$-rooted* graph (with $q \in \mathbb{N}$) is a pair $\mathbf{G} = (G,\mathbb{R})$ where $G$ is a graph and $\mathbb{R}$ is a sequence over a subset $R$ of $q$ vertices of $G$, called *roots*. A *rooted graph* is any $q$-rooted graph, where $q \geq 0$. We treat every graph $G$ as the 0-rooted graph $(G,\langle\rangle)$. The rooted graph $(G,\mathbb{R})$ is *connected* if either $G$ is connected or if every connected component of $G$ contains at least one vertex from $\mathbb{R}$. It is *biconnected* if adding an edge between every pair of root vertices yields a biconnected graph. *Gluing* two $q$-rooted graphs $(G_1,\mathbb{R}_1)$ and $(G_2,\mathbb{R}_2)$ results in the graph $(G_1,\mathbb{R}_1) \oplus (G_2,\mathbb{R}_2)$ obtained by identifying the vertex $\mathbb{R}_1(i)$ with $\mathbb{R}_2(i)$ for every $i \in [q]$. The operation of gluing $k \geq$ copies of a rooted graph $\mathbf{K}$ is denoted by $k \times \mathbf{K}$ and is defined in the obvious way (keep always in mind that the result is a graph). A rooted graph $\mathbf{H} = (H,\mathbb{T})$ is a contraction of a rooted graph $\mathbf{G} = (G,\mathbb{R})$, denoted $\mathbf{H} \preceq \mathbf{G}$ if a rooted graph isomorphic to $(H,\mathbb{T})$ can be obtained after a series of edge contractions on $G$, under the constraint that no path between two vertices of $\mathbb{R}$ can be contracted to a single vertex. If a vertex $v \in V(H)$ results from the contraction of an edge incident to a root vertex of $\mathbb{R}$, then $v$ is a root vertex of $\mathbb{T}$.

**Tree vertex separation.** A *layout* $\sigma$ of a rooted graph $\mathbf{G} = (G,\mathbb{R})$ is a sequence over $V(G)$ such that for every $1 \leq j \leq |\mathbb{R}|$, $\sigma^{-1}(j) \in \mathbb{R}$. We denote by $\mathcal{L}(\mathbf{G})$ the set of all layouts of $\mathbf{G}$. For every $i \in [n]$, the *supporting set* of position $i$ is the set $S_\sigma(i) = \big\{x \in V(G) \mid \sigma(x) < i$ and there exists a $(x,\sigma_i)$-path whose internal vertices belong to $\sigma_{>i}\big\}$. The so-called *tree vertex separation number* of a rooted graph $\mathbf{G}$ is defined as $\mathsf{tvs}(\mathbf{G}) = \min\big\{\mathsf{tcost}(\mathbf{G},\sigma) \mid \sigma \in \mathcal{L}(G)\big\}$, where $\mathsf{tcost}(\mathbf{G},\sigma) = \max\{\big|S_\sigma(i)\big| \mid i \in [n]\}$.

**Search strategies against a lazy robber.** A *search strategy on* a graph $G$ is a sequence $\mathcal{S} = \langle S_1,\ldots,S_r\rangle$, with $r \in \mathbb{N}$, over the sets of subsets of vertices of $V(G)$ where $|S_1| = 1$ and for all $i \in [r-1]$, the symmetric difference of $S_i$ and $S_{i+1}$ has cardinality one. Notice that the difference between two consecutive set either corresponds to a placement or to the removal of a searcher on some vertex $v$. The *cost* of a search strategy $\mathcal{S}$ is $\mathsf{cost}(\mathcal{S}) = \max\{|S_i| \mid i \in [r]\}$.

For a search strategy $\mathcal{S}$ against a lazy robber, we define the *sequence of robber spaces* as the sequence $\mathcal{F}_\mathcal{S} = \langle F_1,\ldots,F_r\rangle$ where:

- $F_1 = V(G) \setminus S_1$.
- For $i \in [2,r]$, let $F_i = (F_{i-1} - S_i) \cup \{v \in V - S_i$ : there is a path from a vertex $u \in F_{i-1} \cap (S_i - S_{i-1})$ to $v$ whose vertices except $u$ belong to $V - S_i\}$.

The complementary sequence $\overline{\mathcal{F}}_\mathcal{S} = \langle \overline{F}_1,\ldots,\overline{F}_r\rangle$ is the sequence of *clean* spaces. We say that the search strategy $\mathcal{S}$ is *complete*, if $F_r = \emptyset$; *monotone*, if for each $i \in [r-1]$, $F_{i+1} \subseteq F_i$. We define $\mathsf{lns}(G)$ as the minimum cost of a complete (or, alternativaly, *cop-win*) monotone search strategy on $G$ against a lazy robber.

## 3 Parameter equivalences

▶ **Proposition 1** ([13, 42]). *For any graph $G$, we have* $\mathsf{tw}(G) = \mathsf{tvs}(G) = \mathsf{lns}(G) - 1$.

To prove a result similar to the above well-known theorem, we adapt the definitions of graph search, tree decomposition, layouts and the associated parameters to the connected setting[3].

- A (monotone and complete) search strategy $\mathcal{S} = \langle S_1, \ldots, S_r \rangle$ of a graph $G$ is *connected* if at every step $i \in [r]$ the clean space $\overline{F}_i$ is connected. We define the parameter $\mathsf{mclns}(G)$ as the minimum cost of a monotone, complete and connected strategy on $G$ against a lazy robber.
- A tree-decomposition $(T, \mathcal{F})$ of a graph $G$ is *connected* if there exists a node $r \in V(T)$ such that for every node $u \in V(T)$, the subgraph $G[V_u]$ is connected, where $V_u$ contains all the vertices that belong to some bag $X_t$ associated with a node $t$ in the $u, r$-path in $T$. We then define the *connected treewidth* $\mathsf{ctw}(G)$ as the minimum width of a connected tree-decomposition. Figure 2 provides an example where the treewidth and the connected treewidth of a graph differs.
- A layout $\sigma$ of a graph $G$ is *connected* if for every $i \in [n]$, the subgraph $G[\sigma_{\leq i}]$ is connected. We let $\mathcal{L}^c(G)$ denote the set of connected layouts of $G$. We then define the *connected tree vertex separation* parameter as $\mathsf{ctvs}(G) = \min\{\mathsf{tcost}(G, \sigma) \mid \sigma \in \mathcal{L}^c(G)\}$.

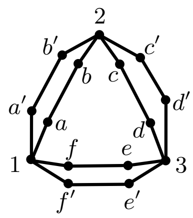


**Figure 2** A series-parallel graph $G$ with $\mathsf{tw}(G) = 2$ and $\mathsf{ctw}(G) = 3$. A connected tree-decomposition of minimum width is given by the path-decomposition $(P, \mathcal{F})$ where $V(P) = \{x_1, \ldots x_8\}$ and $\mathcal{F} = \{X_1 = \{1, a, b, 2\}, X_2 = \{1, a', b', 2\}, X_3 = \{1, 2, c, d\}, X_4 = \{1, 2, d, 3\}, X_5 = \{1, 2, 3, c'\}, X_6 = \{1, 3, c', d'\}, X_7 = \{1, 3, e, f\}, X_8 = \{1, 3, e', f'\}\}$, the root node being $x_1$.

Let us now state the main theorem of this section.

▶ **Theorem 2.** *For every connected graph $G$, we have $\mathsf{ctw}(G) = \mathsf{ctvs}(G) = \mathsf{mclns}(G) - 1$.*

We stress that if in the proof above we use connected path decompositions instead of connected tree decompositions, we obtain the counterpart of Theorem 2 linking the connected path-width of a graph to the connected search number against an agile robber and to a parameter called *connected path vertex separation number*.

## 4 General properties of obstructions

A graph class $\mathcal{G}$ is closed under contraction, if every graph $H$, that is a contraction of a member $G$ of $\mathcal{G}$, also belongs to $\mathcal{G}$. Assume $\mathcal{G}$ is closed under contraction, then a graph $G$ is a *contraction obstruction* to $\mathcal{G}$, if $G \notin \mathcal{G}$ but $H \in \mathcal{G}$ for every $H \preceq G$. Similarly, a graph parameter $\kappa(\cdot)$ is closed under contraction if for every pair of graphs $H$ and $G$ such that $H \preceq G$, $\kappa(H) \leq \kappa(G)$.

[3] These definitions naturally extend to rooted graphs.

The following lemma, stated in terms of rooted graphs, proves that the parameters $\mathsf{ctw}(\cdot)$, $\mathsf{mclns}(\cdot)$ and $\mathsf{ctvs}(\cdot)$ are closed under contraction.

▶ **Lemma 3.** *Let $(G_1, \mathbb{R}_1)$ and $(G_2, \mathbb{R}_2)$ be two $q$-rooted graphs such that $(G_1, \mathbb{R}_1) \preceq (G_2, \mathbb{R}_2)$. Then $\mathsf{ctvs}(G_1, \mathbb{R}_1) \leq \mathsf{ctvs}(G_2, \mathbb{R}_2)$.*

## 4.1 Non-biconnected obstructions

We extend the notion of obstruction sets to rooted graphs in the natural manner. For every $q \geq 1$, we let $\mathcal{O}_k^{(q)}$ denoted the set containing every $q$-rooted graph $\mathbf{G} = (G, \mathbb{R})$, where $\mathsf{ctvs}(\mathbf{G}) > k$ and for every proper contraction $G'$ of $G$, $\mathsf{ctvs}(G', \mathbb{R}') \leq k$. We can prove that an obstruction contains at most one cut vertex, meaning that knowing the set of biconnected obstructions and of 1-rooted obstructions will be enough to describe the full set of obstructions.

We now introduce some concepts on graphs. A vertex subset $S \subseteq V(G)$ is a *separator* if $G \setminus S= G[V \setminus S]$ contains more connected components than $G$. A connected component $H$ of $G \setminus S$ is a *full $S$-component* of $G$ if $N_G(V(H)) = S$. We denote by $\mathcal{C}(G, S)$ the set of all full $S$-connected components of $G$. We denote by $\mathcal{F}(G, S)$ the set containing every induced subgraph $G[S \cup C]$ with $C \in \mathcal{C}(G, S)$. A separator $S$ is a *minimal separator* if $|\mathcal{F}(G, S)| \geq 2$. A minimal separator $S$ is a *minimal $\langle x, y\rangle$-separator* if $x$ and $y$ belong to different full $S$-components. A vertex $x \in V(G)$ is a *cut-vertex* if $\{x\}$ is a separator. The set of cut vertices of a graph $G$ is denoted $C(G)$. A graph $G$ is *biconnected* if it is connected and $C(G) = \emptyset$. A *biconnected component* of a graph is any biconnected subgraph of $G$ that is vertex-maximal. Let $x \in C(G)$ be a cut vertex of $G$. The pair $(G, x)$ is called a *s-pair*. If $Z \in \mathcal{F}(G, \{x\})$, then the 1-rooted graph $(Z, \langle x\rangle)$ is a 1-*component* of the s-pair $(G, x)$. Similarly, if $\{x, y\}$ is a minimal separator of $G$, then the triple $(G, x, y)$ is called a *s-triple*. A 2-rooted graph $(H, \langle x, y\rangle)$ is a 2-*component* of the s-triple $(G, x, y)$ if $\{x, y\}$ is a minimal separator of $G$ and $H \in \mathcal{F}(G, \{x, y\})$.

A vertex $v$ of a graph $G$ is called *$k$-simplicial* if it has degree at most $k$ and its neighborhood induces a complete subgraph. The proof of the next lemma is presented in Section 4 of the (attached) full version.

▶ **Lemma 4.** *For every $k \geq 1$ and every connected graph $G$, $G \in \mathcal{O}_k$ is not biconnected iff $G$ contains exactly one cut vertex and $G \in \{\boldsymbol{A} \oplus \boldsymbol{B} \mid \boldsymbol{A}, \boldsymbol{B} \in \mathcal{O}_k^{(1)}\}$.*

The proof of Lemma 4 is a consequence of Lemma 3 and the next two Lemmas .

▶ **Lemma 5.** *If a connected graph $G$ contains a $k$-simplicial vertex $v$, then $G \notin \mathcal{O}_k$.*

**Proof.** (sketch) The argument simply follows from the observation that $G' = G - v$ is a contraction of $G$ and that extending a connected layout $\sigma'$ of $G'$ by adding $v$ as the last vertex yields a connected layout $\sigma$ of $G$ such that $\mathsf{tcost}(G, \sigma) = \mathsf{tcost}(G', \sigma')$. ◀

▶ **Lemma 6.** *Let $G$ be a connected graph. If $G \in \mathcal{O}_k$ and contains a cut vertex $v$, then the s-pair $(G, v)$ contains exactly two 1-components and $v$ is the unique cut vertex of $G$.*

**Proof.** (sketch) Suppose that $v$ is a cut vertex of $G \in \mathcal{O}_k$ and that $C_0, C_1, C_2$ are distinct connected components of the graph $G - v$. It follows from Lemma 3 that for every $i \in \{0, 1, 2\}, \mathsf{ctvs}(G[C_i \cup C_{(i+1) \bmod 3} \cup \{v\}]) \leqslant k$, which implies that for every $i \in \{0, 1, 2\}$, $\mathsf{ctvs}(G[C_i \cup \{v\}], \{v\}) \leq k$ or $\mathsf{ctvs}(G[C_{(i+1) \bmod 3} \cup \{v\}], \{v\}) \leq k$. Using the connected layouts that certifies these later inequalities, one can build a connected layout $\sigma$ of $G$ such that $\mathsf{tcost}(G, \sigma) \leq k$, a contradiction to the fact that $G \in \mathcal{O}_k$.

So removing a cut vertex in $G$ leaves exactly two connected components. Suppose that there exist two cut vertices $x$ and $y$ and let $C_x$ (resp. $C_y$) be the connected component of $G-x$ not containing $y$ (resp. of $G-y$ not containing $x$). Then applying arguments similar as the ones above to the subgraphs $G_x = [C_x \cup \{x\}]$, $G_y = G[C_y \cup \{y\}]$ and $G_{xy} = G-(C_x \cup C_y)$ allows to show the existence of a connected layout $\sigma$ of $G$ such that $\mathsf{tcost}(G,\sigma) \leq k$, a contradiction to the fact that $G \in \mathcal{O}_k$. ◀

## 4.2 On the price of connectivity

We next examine the question of the price of connectivity for connected treewidth. Let us recall that it is known that the connected pathwidth is a most twice the pathwidth of a graph. Concerning treewidth, as a consequence of Proposition 1 and of the proof of Proposition 7 below, we know that there exists graphs of treewidth at most 4 with abritrary large connected treewidth. Moreover increasing the connected treewidth by one requires to double the number of vertices.

▶ **Proposition 7** ([24]). *For any $n_0$, there is $n \geq n_0$ and an $n$-vertex graph $G$ such that $\mathit{mclns}(G) \in \Omega(\mathit{mlns}(G) \cdot \log n)$.*

We strengthen the theorem above by proving that this result also holds when restricting to series-parallel graphs (that are biconnected graphs of treewidth at most two). Our construction yields to way more simpler graphs than in [24]. The proof of the next result is in Section 6 of the full version.

▶ **Theorem 8** (Corollary 2 in the full version). *For every $k \in \mathbb{N}$, the obstruction set $\mathbf{obs}(\mathcal{T}_k^c)$ contains infinitely many series-parallel graphs.*

To prove Theorem 8, we construct an infinite family of series-parallel graphs with arbitrarily large connected treewidth. For $k \geq 2$, we define the family $\mathcal{Q}_k = \{\mathbf{A} \oplus \mathbf{B} \mid \mathbf{A}, \mathbf{B} \in \mathcal{Y}_k\}$ where $\mathcal{Y}_k$ is the family of 1-rooted graphs $\mathbf{Y}_k = (Y_k, \langle r \rangle)$ that can be constructed as follows: take any tree $T_k$, rooted at vertex $r$, such that the distance between every leaf and $r$ is $k$ and every non-leaf vertex has at least two children; add an *apex* vertex $z$ universal to the leaves of $T_k$; if $r$ has only two neighbors, these neighbors may or may not be adjacent to each another.

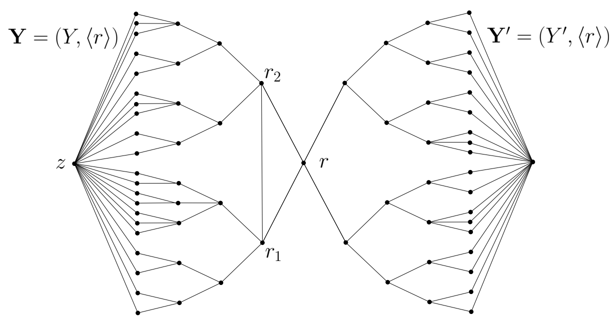


**Figure 3** A graph $H = \mathbf{Y} \oplus \mathbf{Y'} \in \mathcal{O}_4$ with $\mathbf{Y} = (Y, \langle r \rangle) \in \mathcal{Y}_4$ and $\mathbf{Y'} = (Y', \langle r \rangle) \in \mathcal{Y}_4$.

Theorem Theorem 8 is based on the following lemma.

▶ **Lemma 9.** *For every $k \geq 2$, $\mathcal{Q}_k \subseteq \boldsymbol{obs}_{\preceq}(\mathcal{T}_k^c)$.*

**Proof.** (sketch) We first observe that every 1-rooted graph $\mathbf{Y} = (Y, \langle r\rangle) \in \mathcal{Y}_k$ can be constructed from two (or more) graphs $\mathbf{Y}_1 = (Y_1, \langle r_1\rangle) \in \mathcal{Y}_{k-1}$ and $\mathbf{Y}_2 = (Y_2, \langle r_2\rangle) \in \mathcal{Y}_{k-1}$ by identifying their apex vertices and adding a root vertex $r$ adjacent to the roots $r_1$ and $r_2$ of $\mathbf{Y}_1$ and $\mathbf{Y}_2$ (see Figure 3). For any $\mathbf{Y}_k = (Y_k, \langle r\rangle) \in \mathcal{Y}_k$, we define the 2-rooted graphs $\mathbf{Y}_k^{(2)} = (Y, \langle r, z\rangle)$ where $z$ is the apex vertex of $\mathbf{Y}_k$. The proof relies on the following claims, that for every $k \geq 2$: (a) $\mathsf{ctvs}(\mathbf{Y}_k^{(2)}) = k$, (b) $\mathsf{ctvs}(\mathbf{Y}_k) > k$, and (c) for every edge $e$ of $Y_k$, $\mathsf{ctvs}((Y_k/e, \langle r\rangle)) \leq k$.

Let us sketch the argument of the second claim. Consider a connected layout $\sigma \in \mathcal{L}^c(\mathbf{Y}_k)$ and suppose that $\sigma_i = z$. By the connectivity of $\sigma$, the induced subgraph $Y_k[\sigma_{\leq i}]$ contains a path $P$ from the root $r$ to the apex $z$. Observe that $P$ contains exactly $k+2$ vertices $r, v_2, \dots v_{k+1}, z$ and that $Y_k$ contains $k+1$ internally vertex disjoint paths from the apex $z$ to $r, v_2, \dots v_{k+1}$. It follows that the supporting set $S_\sigma(i)$ contains at least $k+1$ vertices.

From the second and the third claims we conclude that every $\mathbf{Y}_k = (Y_k, \langle r\rangle) \in \mathcal{Y}_k$ belongs to the set $\mathcal{O}_k^1$ of 1-rooted obstructions of $\mathcal{T}_k^c$. By Lemma 4, we conclude that $\mathcal{Q}_k \subseteq \mathbf{obs}_{\preceq}(\mathcal{T}_k^c)$. ◀

## 5 The obstruction set $\mathcal{O}_2$

Thanks to Lemma 4, the non-biconnected parts of $\mathcal{O}_2$ can be determined if we identify the 1-rooted obstruction set $\mathcal{O}_2^{(1)}$. To that aim, let us first define the family $\mathcal{B}_2^{(1)} = \{\mathbf{Y}_x\} \cup \{\mathbf{Y}_x^{(k)} \mid k \geq 2]\}$ where $\mathbf{Y}_x^{(k)} = (k \times \mathbf{R}_x^y, \langle x\rangle)$ (see Figure 4). It is not difficult to check that these graphs are 1-rooted obstructions.

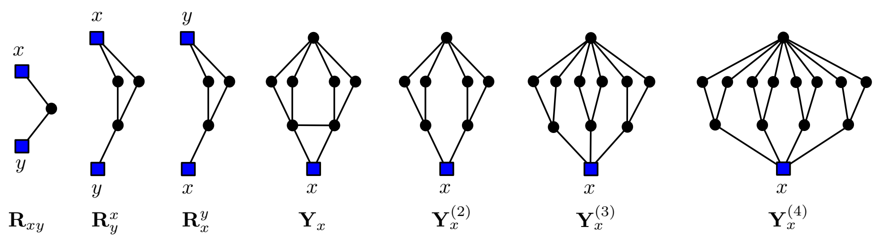


**Figure 4** The rooted graphs $\mathbf{R}_{xy}$, $\mathbf{R}_x^y$, $\mathbf{R}_y^x$, $\mathbf{Y}_x$, $\mathbf{Y}_x^{(2)}$, $\mathbf{Y}_x^{(3)}$, and $\mathbf{Y}_x^{(4)}$.

It can be easily checked that the three biconnected graphs depicted in Figure 5 belong to $\mathcal{O}_2$. We define the set $\mathcal{B}_2 = \{K_4, W_1, W_2\} \cup \{\mathbf{A} \oplus \mathbf{B} \mid \mathbf{A}, \mathbf{B} \in \mathcal{B}_2^{(1)}\}$.

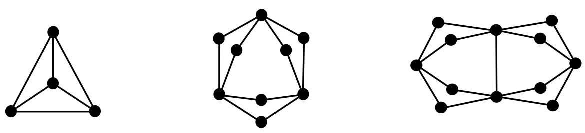

**Figure 5** From left to right, the graphs $K_4, W_1$, and $W_2$.

**2-twin expansion.** Let $\mathbf{G} = (G, \mathbb{R})$ be a rooted graph and let $S \subseteq V(G)$. We say that $S$ is a 2-*twin family* of $\mathbf{G}$ if $S \cap V(\mathbb{R}) = \emptyset$, $|S| \geq 2$ and there are two vertices $a, b \in V(G)$ such that $\forall s \in S, N_G(s) = \{a, b\}$. We call the vertices $a, b$ the *bases* of the 2-twin family $S$. We say that a graph $\mathbf{G'} = (G', \mathbb{R}')$ is a 2-*twin expansion of* $G$ if $\mathbb{R} = \mathbb{R}'$ and $G'$ is obtained from $G$ by adding vertices such that each additional vertex is made adjacent with the base vertices of some of the 2-twin families of $\mathbf{G}$. Given a class of rooted graphs $\mathcal{C}$ we define its 2-*twin expansion* $\mathsf{texp}(\mathcal{C})$ as the class of rooted graphs containing all 2-twin expansions of all the graphs in $\mathcal{C}$. We are now ready to state the main result of this section.

▶ **Theorem 10.** *A graph $G$ belongs to $\mathcal{T}_2^c$ if and only if it does not contains a graph of* $\mathsf{texp}(\mathcal{B}_2)$ *as a contraction, that is* $\mathcal{O}_2 = \mathsf{texp}(\mathcal{B}_2)$.

## 5.1 Some elements of the proof of Theorem 10

**The set $\mathcal{O}_2$ is closed under twin expansion.** We say that a rooted graph $\mathbf{G}$ is *simplified* if all its 2-twin families have size 2. Given a rooted graph $\mathbf{G}$ we denote by $\tilde{\mathbf{G}}$ the unique simplified rooted graph such that $\mathbf{G} \in \mathsf{texp}(\{\tilde{\mathbf{G}}\})$. Given a set $\mathcal{C}$ of rooted graphs, we define $\tilde{\mathcal{C}} = \{\tilde{\mathbf{G}} \mid \mathbf{G} \in \mathcal{C}\}$. Observe that every graph of $\mathcal{B}_2$ is simplified.

▶ **Lemma 11.** *A graph $G$ belongs to $\mathcal{O}_2$ iff $\tilde{G}$ belongs to $\mathcal{O}_2$.*

The next lemma is the extension of Lemma 11 to 1-rooted obstructions. Its proof mainly follows from Lemma 4.

▶ **Lemma 12.** *Let $\mathbf{H} = (H, \langle v\rangle)$ be a 1-rooted graph. Then $\mathbf{H} \in \mathcal{O}_2^{(1)}$ if and only if $\tilde{\mathbf{H}} \in \tilde{\mathcal{O}}_2^{(1)}$.*

**Simplified obstructions.** We now identify sets of simplified graphs, 1-rooted graphs and 2-rooted graphs that are obstructions. Later, we prove that from these sets the full set of obstructions to $\mathcal{T}_2^c$ can be constructed. The following Lemmas establish that the sets $\mathcal{B}_2^{(1)}$ and $\mathcal{B}_2$ build from the graphs of Figure 4 and Figure 5 are simplified obstructions.

▶ **Lemma 13.** *If a 1-rooted graph $\boldsymbol{G} = (G, \langle x\rangle)$ belongs to $\mathsf{texp}(\mathcal{B}_2^{(1)})$, then $\boldsymbol{G} \in \mathcal{O}_2^{(1)}$. If a graph $G$ belongs to $\mathsf{texp}(\mathcal{B}_2)$, then $G$ belongs to $\mathcal{O}_2$.*

Let us now turn to biconnected 2-rooted obstructions. We define the set $\mathcal{B}_2^{(2)} = \{\mathbf{R}^{xy}, \mathbf{R}^{xy+}, \mathbf{K}_4^{xy-}, \mathbf{K}_4^{xy}\}$ of 2-rooted graphs depicted in Figure 6. We say that a biconnected 2-rooted graph $\mathbf{H} = (H, \langle x, y\rangle)$ is *elementary* if it is $\mathsf{texp}(\mathcal{B}_2^{(2)})$-free.

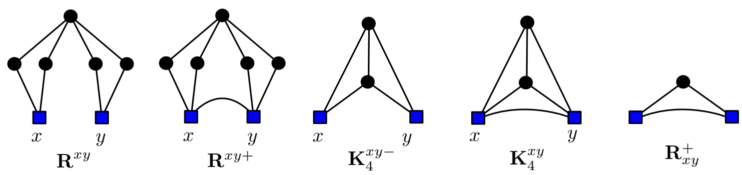


**Figure 6** The 2-rooted graphs $\mathbf{R}^{xy}, \mathbf{R}^{xy+}, \mathbf{K}_4^{xy-}, \mathbf{K}_4^{xy}$ and $\mathbf{R}_{xy}^+$.

▶ **Lemma 14.** *The set of biconnected graphs in $\mathcal{O}_2^{(2)}$ is $\mathsf{texp}(\mathcal{B}_2^{(2)})$.*

**Proof.** (sketch) By considering a minimal counter-example, the proof first establishes that every elementary biconnected 2-rooted graph belongs to $\mathcal{T}_2^c$. Then we check that for every 2-rooted graph $\mathbf{G} \in \mathsf{texp}(\mathcal{B}_2^{(2)})$, $\mathsf{ctw}(\mathbf{G}) \geq 3$ but every contraction of $\mathbf{G}$ belongs to $\mathcal{T}_2^c$. ◀

**Structure of obstructions.** Let $xy$ be an edge of a graph $G$. We say that $xy$ is a *separating edge* if the set $\{x, y\}$ is a minimal separator. We say that $xy$ is a *marginal edge* if there is a vertex $z$ such that both $(G, x, z)$ and $(G, y, z)$ are s-triples.

▶ **Lemma 15.** *Let $G$ be a graph in $\tilde{\mathcal{O}}_2$. If $G$ contains a separating edge $xy$, then either $G$ is isomorphic to $W_2$ or $G$ contains a cut-vertex $r$ and the 1-component of the s-pair $(G, r)$ containing $xy$ is isomorphic to $\mathbf{Y}_r$.*

▶ **Lemma 16.** *Let $G \in \tilde{\mathcal{O}}_2$ be a graph without separating edge. If $(G, x, y)$ is a s-triple, then*
1. *either every 2-component of $(G, x, y)$ is elementary,*
2. *or there exists a non-elementary 2-component of $(G, x, y)$, denoted by $\boldsymbol{H} = (H, \langle x, y\rangle)$, such that $G \setminus (V(H) \setminus \{x, y\})$ cannot be contracted to $\ell \times \boldsymbol{R}_{xy}$ for any $\ell \geq 2$.*

From Lemma 15 and Lemma 16, we deduce a series of properties needed to understand the role of marginal edges and to conclude the characterization of $\mathcal{O}_2$.

▶ **Lemma 17.** *Let $(G, x, y)$ be an s-triple of $G \in \tilde{\mathcal{O}}_2$. If $(H, \langle x, y\rangle)$ is a 2-component of $(G, x, y)$ that is isomorphic to $\boldsymbol{R}_x^y$, then $x$ is a cut-vertex.*

▶ **Lemma 18.** *Let $(G, x, y)$ be an s-triple of $G \in \tilde{\mathcal{O}}_2$. If $\boldsymbol{H}$ is an elementary 2-component of $(G, x, y)$ without cut-vertex, then $\boldsymbol{H}$ is isomorphic to $\boldsymbol{R}_{xy}$.*

▶ **Lemma 19.** *Let $\boldsymbol{G} = (G, \langle x\rangle) \in \tilde{\mathcal{O}}_2^{(1)}$.*
1. *If $(G, x, y)$ is a s-triple, then none of its 2-components is isomorphic to $\boldsymbol{R}_{xy}$.*
2. *If $\boldsymbol{H} = (H, \langle x, y\rangle)$ is an elementary 2-component of an s-triple $(G, x, y)$, then $\boldsymbol{H}$ is isomorphic to $\boldsymbol{R}_x^y$.*

**Biconnected obstructions.** We now have all the ingredients for the proof of Theorem 10. We start with the identification of the biconnected elements of $\tilde{\mathcal{O}}_2$.

▶ **Lemma 20.** *No biconnected graph in $\tilde{\mathcal{O}}_2$ contains a marginal edge.*

▶ **Lemma 21.** *The biconnected graphs in $\tilde{\mathcal{O}}_2$ are the graphs $K_4$, $W_1$, and $W_2$.*

**Proof.** (sketch) For a contradiction, we suppose that $\tilde{\mathcal{O}}_2$ contains a graph $G$ distinct from $K_4$, $W_1$, and $W_2$. From Lemma 15, $G$ does not contain a separating edge. As it excludes $K_4$ as a contraction, it contains a degree-two vertex $a$. Let $x$ and $y$ be the two neighbors of $a$ and let $\mathcal{H} = \{\mathbf{H}_0, \ldots, \mathbf{H}_q\}$ be the 2-components of the s-triple $(G, x, y)$ with $V(H_0) = \{a, x, y\}$. As $G$ is biconnected, so is every 2-rooted graph in $\mathcal{H}$. We next prove that exactly one of the 2-rooted graphs in $\{\mathbf{H}_1, \ldots, \mathbf{H}_q\}$, say $\mathbf{H}_1$, is not elementary. Then by Lemma 18, every $\mathbf{H}_j \in \mathcal{H}$ distinct from $\mathbf{H}_1$ is isomorphic to $\mathbf{R}_{xy}$. As $G$ is simplified, we have $q \leq 2$. If $q = 2$, as $\mathbf{H}_0 \oplus \mathbf{H}_2 = 2 \times \mathbf{R}_{xy}$, as $\mathbf{H}_1$ is not elementary and as $G$ does not contains a separating edge, Lemma 16 leads to a contradiction. In the case $q = 1$, it can be proved that $H_1$ contains a cut vertex, implying the existence of a marginal edge in $G$, a contradiction to Lemma 20. ◀

**Non-biconnected obstructions.** The second part of the proof of Theorem 10 identifies the non-biconnected elements of $\tilde{\mathcal{O}}_2$.

▶ **Lemma 22.** *The non-biconnected graphs in $\tilde{\mathcal{O}}_2$ are the graphs in $\{\boldsymbol{A} \oplus \boldsymbol{B} \mid \boldsymbol{A}, \boldsymbol{B} \in \mathcal{B}_2^{(1)}\}$.*

**Proof.** (sketch) From Lemma 4 and Lemma 13, it is enough to prove that $\tilde{\mathcal{O}}_2^{(1)} \subseteq \mathcal{B}_2^{(1)}$. We assume, towards a contradiction, that there is some 1-rooted graph $\mathbf{G} = (G, \langle r\rangle) \in \tilde{\mathcal{O}}_2^{(1)} \setminus \mathcal{B}_2^{(1)}$. Observe that $\mathbf{G}$ is $\mathcal{B}_2^{(1)}$-free and $G$ is biconnected. From Lemma 15, we can assume that $G$ does not have separating edges. Let $J = 2 \times \mathbf{G}$. As the underlying graphs of the 2-rooted graphs in $\mathcal{B}_2^{(1)}$ are $\{K_4, W_1, W_2\}$-free, Lemma 4 implies that $J$ is $\mathcal{B}_2$-free and thereby $K_4$-free. It can easily be seen that $r$ has more than two neighbors. Also one may consider a 2-tree $T$ that contains $G$ as a spanning subgraph and satisfies the following properties

- (D1) If an edge is marginal in $T$ then it is also marginal in $G$.
- (D2) If an edge is simplicial in $T$ then one of its endpoints have degree 2 in $G$.
- (D3) If an edge is a separating edge of $G$, then it is also a separating edge in $T$.

Let $z$ be a neighbor of $r$. Because of (D3), the edge $e = rz$ is either a marginal or a simplicial edge of $T$. We claim that $e$ is marginal. Indeed, if $e$ is simplicial, then from (D2) $z$ has degree 2. Let $w$ be the other neighbor of $z$. Notice that one of the 2-components of the s-triple $(G, r, w)$ is isomorphic to $\mathbf{R}_{rw}$, a contradition to Lemma 19. We now know that $e = rz$ is a marginal edge. Let $t$ be the base of $e$. cleanly $(G, r, t)$ is an $s$-triple and $tr \notin E(G)$ as $G$ does not have separating edges. We denote by $\mathcal{U} = \mathbf{U}_1, \ldots, \mathbf{U}_q$ the 2-components of $(G, r, t)$. Our next step is to prove that all 2-rooted graphs in $\mathcal{U}$ are simple. This, together with Lemma 19 imply that all graphs in $\mathcal{U}$ are isomorphic to $\mathbf{R}_r^t$. This means that $\mathbf{G}$ contains as a contraction some $\mathbf{Y}_t^{(\ell)}$ for some $\ell \geq 3$. As each such $\mathbf{Y}_t^{(\ell)}$ belongs to $\mathcal{B}_2^{(1)}$ we have a contradiction. ◀

## References


1 Isolde Adler. Open Problems related to computing obstruction sets. Manuscript, September 2008.

2 Steve Alpern and Shmuel Gal. *The theory of search games and rendezvous*. International Series in Operations Research & Management Science, 55. Kluwer Academic Publishers, Boston, MA, 2003.

3 Brian Alspach. Searching and sweeping graphs: a brief survey. *Matematiche (Catania)*, 59(1-2):5–37 (2006), 2004.

4 Stefan Arnborg, Andrzej Proskurowski, and Derek G. Corneil. Forbidden minors characterization of partial 3-trees. *Discrete Mathematics*, 80(1):1–19, 1990.

5 Lali Barrière, Paola Flocchini, Fedor V. Fomin, Pierre Fraigniaud, Nicolas Nisse, Nicola Santoro, and Dimitrios M. Thilikos. Connected graph searching. *Inf. Comput.*, 219:1–16, 2012. doi:10.1016/j.ic.2012.08.004.

6 Lali Barrière, Pierre Fraigniaud, Nicola Santoro, and Dimitrios M. Thilikos. Searching Is Not Jumping. In *29th International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2003)*, volume 2880 of *LNCS*, pages 34–45. Springer, 2003.

7 Micah J Best, Arvind Gupta, Dimitrios M. Thilikos, and Dimitris Zoros. Contraction obstructions for connected graph searching. *Discrete Applied Mathematics*, 209:27–47, 2016. 9th International Colloquium on Graph Theory and Combinatorics, 2014, Grenoble.

8 D. Bienstock and Paul Seymour. Monotonicity in graph searching. *J. Algorithms*, 12(2):239–245, 1991.

9 Dan Bienstock, Neil Robertson, Paul D. Seymour, and Robin Thomas. Quickly excluding a forest. *J. Comb. Theory Ser. B*, 52(2):274–283, 1991.

10 Daniel Bienstock. Graph searching, path-width, tree-width and related problems (a survey). In *Reliability of computer and communication networks (New Brunswick, NJ, 1989)*, volume 5 of *DIMACS Ser. Discrete Math. Theoret. Comput. Sci.*, pages 33–49. Amer. Math. Soc., Providence, RI, 1991.

11 R. Breisch. An intuitive approach to speleotopology. *Southwestern Cavers (A publication of the Southwestern Region of the National Speleological Society)*, VI(5):72–78, 1967.

12 Erik D. Demaine, MohammadTaghi Hajiaghayi, and Ken-ichi Kawarabayashi. Algorithmic Graph Minor Theory: Improved Grid Minor Bounds and Wagner's Contraction. *Algorithmica*, 54(2):142–180, 2009. doi:10.1007/s00453-007-9138-y.

13 Nick D. Dendris, Lefteris M. Kirousis, and Dimitrios M. Thilikos. Fugitive-search games on graphs and related parameters. *Theoret. Comput. Sci.*, 172(1-2):233–254, 1997.

14 Dariusz Dereniowski. From Pathwidth to Connected Pathwidth. *SIAM J. Discrete Math.*, 26(4):1709–1732, 2012. doi:10.1137/110826424.

15 Reinhard Diestel and Malte Müller. Connected Tree-Width. *Combinatorica*, 38(2):381–398, 2018. doi:10.1007/s00493-016-3516-5.

16 Paola Flocchini, Miao Jun Huang, and Flaminia L. Luccio. Contiguous Search in the Hypercube for Capturing an Intruder. In *Proceedings of the 19th International Parallel and Distributed Processing Symposium (IPDPS 2005) IPDPS*. IEEE Computer Society, 2005.

17 Paola Flocchini, Miao Jun Huang, and Flaminia L. Luccio. Decontamination of chordal rings and tori using mobile agents. *Int. J. of Found. of Comp. Sc.*, 18(3):547–564, 2007.

18 Paola Flocchini, Miao Jun Huang, and Flaminia L. Luccio. Decontamination of hypercubes by mobile agents. *Networks*, page to appear, 2007.

19 F. V. Fomin, D. M. Thilikos, and I. Todinca. Connected Graph Searching in Outerplanar Graphs. *Electronic Notes in Discrete Mathematics*, 22:213–216, 2005. 7th International Colloquium on Graph Theory. Short communication.

20 Fedor V. Fomin and Dimitrios M. Thilikos. On the monotonicity of games generated by symmetric submodular functions. *Discrete Appl. Math.*, 131(2):323–335, 2003. Submodularity. `doi:10.1016/S0166-218X(02)00459-6`.

21 Fedor V. Fomin and Dimitrios M. Thilikos. An annotated bibliography on guaranteed graph searching. *Theoret. Comput. Sci.*, 399(3):236–245, 2008.

22 Pierre Fraigniaud and Nicolas Nisse. Connected Treewidth and Connected Graph Searching. In *Proceedings of the 7th Latin American Symposium on Theoretical Informatics (LATIN 2006)*, volume 3887 of *LNCS*, pages 479–490. Springer, 2006.

23 Pierre Fraigniaud and Nicolas Nisse. Connected Treewidth and Connected Graph Searching. In *7th Latin American Symposium on Theoretical Informatics (LATIN 2006)*, volume 3887 of *LNCS*, pages 479–490. Springer, 2006.

24 Pierre Fraigniaud and Nicolas Nisse. Monotony Properties of Connected Visible Graph Searching. In *32nd International Workshop on Graph-Theoretic Concepts in Computer Science (WG 2006)*, volume 4271 of *LNCS*, pages 229–240. Springer, 2006.

25 Matthias Hamann and Daniel Weißauer. Bounding Connected Tree-Width. *SIAM J. Discrete Math.*, 30(3):1391–1400, 2016. `doi:10.1137/15M1044618`.

26 Philippe Jégou and Cyril Terrioux. Bag-Connected Tree-Width: A New Parameter for Graph Decomposition. In *International Symposium on Artificial Intelligence and Mathematics, ISAIM 2014, Fort Lauderdale, FL, USA, January 6-8, 2014*, 2014. URL: `http://www.cs.uic.edu/pub/Isaim2014/WebPreferences/ISAIM2014_Jegou_Terrioux_New.pdf`.

27 Philippe Jégou and Cyril Terrioux. Tree-Decompositions with Connected Clusters for Solving Constraint Networks. In Barry O'Sullivan, editor, *Principles and Practice of Constraint Programming - 20th International Conference, CP 2014, Lyon, France, September 8-12, 2014. Proceedings*, volume 8656 of *Lecture Notes in Computer Science*, pages 407–423. Springer, 2014. `doi:10.1007/978-3-319-10428-7_31`.

28 Philippe Jégou and Cyril Terrioux. Combining restarts, nogoods and bag-connected decompositions for solving CSPs. *Constraints*, 22(2):191–229, 2017. `doi:10.1007/s10601-016-9248-8`.

29 Marcin Kaminski, Daniël Paulusma, and Dimitrios M. Thilikos. Contracting planar graphs to contractions of triangulations. *J. Discrete Algorithms*, 9(3):299–306, 2011. `doi:10.1016/j.jda.2011.03.010`.

30 Nancy G. Kinnersley and Michael A. Langston. Obstruction set Isolation for the Gate Matrix Layout Problem. *Discrete Applied Mathematics*, 54:169–213, 1994.

31 Lefteris M. Kirousis and Christos H. Papadimitriou. Interval graphs and searching. *Discrete Math.*, 55(2):181–184, 1985.

32 Lefteris M. Kirousis and Christos H. Papadimitriou. Searching and pebbling. *Theoret. Comput. Sci.*, 47(2):205–218, 1986.

33 J. Lagergren. Upper bounds on the size of obstructions and intertwines. *J. Comb. Theory, Ser. B*, 73:7–40, 1998.

34 Andrea S. LaPaugh. Recontamination does not help to search a graph. *J. Assoc. Comput. Mach.*, 40(2):224–245, 1993.

35 Thomas W. Mattman. Forbidden Minors: Finding the Finite Few. In Aaron Wootton, Valerie Peterson, and Christopher Lee, editors, *A Primer for Undergraduate Research: From Groups and Tiles to Frames and Vaccines*, pages 85–97, Cham, 2017. Springer International Publishing. doi:10.1007/978-3-319-66065-3_4.

36 Rolf H. Möhring. Graph problems related to gate matrix layout and PLA folding. In *Computational graph theory*, volume 7 of *Comput. Suppl.*, pages 17–51. Springer, Vienna, 1990.

37 Nicolas Nisse. Connected graph searching in chordal graphs. *Discrete Applied Mathematics*, 157(12):2603–2610, 2009. Second Workshop on Graph Classes, Optimization, and Width Parameters. doi:10.1016/j.dam.2008.08.007.

38 Nicolas Nisse. *Network Decontamination*, pages 516–548. Springer International Publishing, Cham, 2019. doi:10.1007/978-3-030-11072-7_19.

39 T. D. Parsons. Pursuit-evasion in a graph. In *Theory and applications of graphs*, volume 642 of *Lecture Notes in Math.*, pages 426–441. Springer, Berlin, 1978.

40 Neil Robertson and P. D. Seymour. Graph Minors. XX. Wagner's conjecture. *Journal of Combinatorial Theory, Series B*, 92(2):325–357, 2004.

41 Daniel P. Sanders. On linear recognition of tree-width at most four. *SIAM J. Discrete Math.*, 9(1):101–117, 1996.

42 P. D. Seymour and Robin Thomas. Graph searching and a min-max theorem for tree-width. *J. Comb. Theory Ser. B*, 58(1):22–33, 1993.