

# Fixed-Parameter Algorithms for Computing RAC Drawings of Graphs

Cornelius Brand<sup>[0000–0002–1929–055X]</sup>, Robert Ganian<sup>[0000–0002–7762–8045]</sup>,  
Sebastian Röder, and Florian Schager

Algorithms and Complexity Group, TU Wien (Vienna University of Technology)  
{cbrand, rganian}@ac.tuwien.ac.at, sebastian.roeder@student.tuwien.ac.at,  
florian.schager@tuwien.ac.at

**Abstract.** In a right-angle crossing (RAC) drawing of a graph, each edge is represented as a polyline and edge crossings must occur at an angle of exactly  $90^\circ$ , where the number of bends on such polylines is typically restricted in some way. While structural and topological properties of RAC drawings have been the focus of extensive research, little was known about the boundaries of tractability for computing such drawings. In this paper, we initiate the study of RAC drawings from the viewpoint of parameterized complexity. In particular, we establish that computing a RAC drawing of an input graph  $G$  with at most  $b$  bends (or determining that none exists) is fixed-parameter tractable parameterized by either the feedback edge number of  $G$ , or  $b$  plus the vertex cover number of  $G$ .

**Keywords:** RAC drawings · fixed-parameter tractability · vertex cover number · feedback edge number

## 1 Introduction

Today we have access to a wealth of approaches and tools that can be used to draw planar graphs, including, e.g., Fáry’s Theorem [29] which guarantees the existence of a planar straight-line drawing for every planar graph and the classical algorithm of Fraysseix, Pach and Pollack [28] that allows us to obtain straight-line planar drawings on an integer grid of quadratic size. However, much less is known about the kinds of drawings that can be achieved for non-planar graphs. The study of combinatorial and algorithmic aspects of such drawings lies at the heart of a research direction informally referred to as “beyond planarity” (see, e.g., the relevant survey and book chapter [21,18]).

An obvious goal when attempting to visualize non-planar graphs would be to obtain a drawing which minimizes the total number of crossings. This question is widely studied within the context of the crossing number of graphs, and while obtaining such a drawing is NP-hard [33] it is known to be fixed-parameter tractable when parameterized by the total number of crossings required thanks to a seminal result of Grohe [34]. However, research over the past twenty years has shown that drawings which minimize the total number of crossings are not necessarily optimal in terms of human readability. Indeed, the topological and

geometric properties of such drawings may have a significantly larger impact than the total number of crossings, as was observed, e.g., by the initial informal experiment of Mutzel [41] and the pioneering set of user experiments carried out by the graph drawing research lab at the University of Sydney [36,38,37]. The latter works demonstrated that “large-angle drawings” (where edge crossings have larger angles) are significantly easier to read than drawings where crossings occur at acute angles.

Motivated by these findings, in 2011 Didimo, Eades, and Liotta investigated graph drawings where edge crossings are only permitted at  $90^\circ$  angles [20] (see Figure 1 for an illustration). Today, these *right-angle crossing* (or *RAC*) drawings are among the best known and most widely studied beyond-planar drawing styles [21,18], with the bulk of the research to date focusing on understanding necessary and sufficient conditions for the existence of such drawings as well as the space they require [3,4,16,17,8,1,2,27]. A prominent theme in the context of RAC drawings concerns the number of times edges are allowed to be bent: it has been shown that every graph admits a RAC drawing if each edge can be bent 3 times [20], and past works have considered straight-line RAC drawings as well as RAC drawings where the number of bends per edge is limited to 1 or 2.

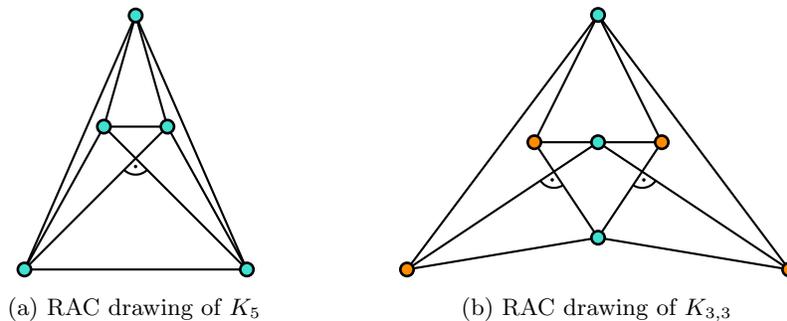


Fig. 1: Examples of RAC drawings.

And yet, in spite of the considerable body of work concentrating on combinatorial and topological properties of such drawings, so far almost nothing is known about the complexity of computing a RAC drawing of a given graph. Indeed, while the problem of determining whether a graph admits a straight-line RAC drawing is NP-hard [4] and was recently shown to be  $\exists\mathbb{R}$ -complete [44], there is a surprising lack of known algorithms that can compute such drawings for special classes of graphs or, more generally, parameterized algorithms that exploit quantifiable properties of the input graph to guarantee the tractability of computing RAC drawings (either without or with limited bends). This gap in our understanding starkly contrasts the situation for so-called 1-planar drawings—another prominent beyond-planar drawing style for which a number of fixed-parameter algorithms are known [6,25,24]—as well as recent advances mapping the boundaries of tractability for other graph drawing problems [35,9,10].

**Contribution.** We initiate an investigation of the parameterized complexity of determining whether a graph  $G$  admits a RAC drawing. Given the well-motivated focus of previous works on limiting the amount of bends in such drawings, an obvious first choice for a parameterization would be to consider an upper bound  $b$  on the total number of bends permitted in the drawing. However, on its own such a parameter cannot suffice to achieve fixed-parameter tractability in view of the NP-hardness of the problem for  $b = 0$ , i.e., for straight-line RAC drawings.

Hence, we turn towards identifying structural parameters of  $G$  that guarantee fixed-parameter RAC drawing algorithms. While established decompositional parameters such as treewidth [43] and clique-width [14] represent natural choices of parameterizations for purely combinatorial problems, the applicability of these parameters in solving graph drawing problems is complicated by the inherent difficulty of performing dynamic programming when the task is to obtain a drawing of the graph. This is why the parameters often used in this setting are non-decompositional, with the most notable examples being the *vertex cover number*  $\mathbf{vcn}$  (i.e., the size of a minimum vertex cover) and the *feedback edge number*  $\mathbf{fen}$  (i.e., the edge deletion distance to acyclicity); further details are available in the overview of related work below. As our main contributions, we provide two novel parameterized algorithms:

1. a fixed-parameter algorithm for determining whether  $G$  admits a RAC drawing with at most  $b$  bends when parameterized by  $\mathbf{fen}(G)$ ;
2. a fixed-parameter algorithm for determining whether  $G$  admits a RAC drawing with at most  $b$  bends when parameterized by  $\mathbf{vcn}(G) + b$ ;

Both of the presented algorithms are constructive, meaning that they can also output a RAC drawing of the graph if one exists. The core underlying technique used in both proofs is that of *kernelization*, which relies on defining reduction rules that can provably reduce the size of the instance until it is upper-bounded by a function of the parameter alone. While kernelization is a well-established and generic technique, its use here requires non-trivial insights into the structural properties of optimal solutions in order to carefully identify parts of the graph which can be simplified without impacting the final outcome.

We prove that both algorithms in fact hold for the more general case where each edge is marked with an upper bound on the number of bends it can support, allowing us to capture the previously studied 1- and 2-bend RAC drawings. Moreover, we show that the latter algorithm can be lifted to establish fixed-parameter tractability when parameterized by  $b$  plus the *neighborhood diversity* (i.e., the number of maximal modules) of  $G$  [40,30,39]. In the concluding remarks, we also discuss possible extensions towards more general parameterizations and apparent obstacles on the way to such results.

**Related Work.** Didimo, Eades and Liotta initiated the study of RAC drawings by analyzing the interplay between the number of bends per edge and the total number of edges [20]. Follow-up works also considered extensions and variants of the initial concept, such as upward RAC drawings [3], 2-layer RAC drawings [16,17] and 1-planar RAC drawings [8]. More recent works investigated

the existence of RAC drawings for bounded-degree graphs [2], and RAC drawings with at most one bend per edge [1]. It is known that every graph admits a RAC drawing with at most three bends per edge [20], and that determining whether a graph admits a RAC drawing with zero bends per edge is NP-hard [4].

The vertex cover number has been used as a structural graph parameter to tackle a range of difficult problems in graph drawing as well as other areas. Fixed-parameter algorithms for drawing problems based on the vertex cover number are known for, e.g., computing the obstacle number of a graph [5], computing the stack and queue numbers of graphs [9,10], computing the crossing number of a graph [35] and 1-planarity testing [6]. Similarly, the feedback edge number (sometimes called the *cyclomatic number*) has been used to tackle problems which are not known to be tractable w.r.t. treewidth, including 1-planarity testing [6] and the EDGE DISJOINT PATHS problem [32] (see also Table 1 in [31]).

These two parameterizations are incomparable: there are problems which remain NP-hard on graphs of constant vertex cover number while being FPT when parameterized by the feedback edge number (such as EDGE DISJOINT PATHS [26,32]), and vice-versa. That being said, the existence of a fixed-parameter algorithm parameterized by the feedback edge number is open for a number of graph drawing problems that are known to be FPT w.r.t. the vertex cover number; examples include computing the aforementioned stack, queue and obstacle numbers.

## 2 Preliminaries

We assume familiarity with standard concepts in graph theory [22]. All graphs considered in this manuscript are assumed to be simple and undirected.

**RAC Drawings.** Given a graph  $G = (V, E)$  on  $n$  vertices with  $m$  edges, a *drawing* of  $G$  is a mapping  $\delta$  that takes vertices  $V$  to points in the Euclidean plane  $\mathbb{R}^2$ , and assigns to every edge  $e = uv \in E$  the image of a simple plane curve  $[0, 1] \rightarrow \mathbb{R}^2$  connecting the points  $\delta(u), \delta(v)$  corresponding to  $u$  and  $v$ . We require that  $\delta$  is injective on  $V$ , and furthermore that for all vertices  $v$  and edges  $e$  not incident to  $v$ , the point  $\delta(v)$  is not contained in  $\text{int}(\delta(e))$ , where  $\text{int}(\delta(e))$  is the image of  $(0, 1)$  under  $\delta$ .

A *polyline drawing* of  $G$  is a drawing such that for each edge  $e \in E$ ,  $\delta(e)$  can be written as a union  $\delta(e) = \lambda_1^e \cup \dots \cup \lambda_t^e$  of closed straight-line segments  $\lambda_1^e, \dots, \lambda_t^e$  such that:

- for each  $1 \leq i \leq t - 1$ , the segments  $\lambda_i^e$  and  $\lambda_{i+1}^e$  intersect in precisely one of their shared end-points and moreover close an angle different than  $180^\circ$ , and
- every other pair of segments is disjoint.

The shared intersection points between consecutive segments are called the *bends* of  $e$  in the drawing  $\delta$ .

For two edges  $e$  and  $f$ , their set of *crossings* in the drawing  $\delta$  is the set  $\text{int}(\delta(e)) \cap \text{int}(\delta(f))$ . We will assume without loss of generality that any drawing  $\delta$  of  $G$  has a finite number of crossings.

The central type of drawing studied in this paper are those that allow only *right-angle crossings* between edge drawings (so-called *RAC drawings*): We say that the edges  $e, f \in E$  have a *right-angle crossing* in a polyline drawing  $\delta$  of  $G$  if the crossing lies in the relative interiors of the respective line segments defining  $\delta(e)$  and  $\delta(f)$ , and most crucially, the intersecting line segments of  $\delta(e)$  and  $\delta(f)$  are orthogonal to each other (i.e., they meet at a right angle). Let  $\delta$  be a polyline-drawing of a graph,  $\beta : E \mapsto \{0, 1, 2, 3\}$  a mapping, and  $b \in \mathbb{N}$  a number. If every crossing of  $\delta$  is a right-angle crossing, the number of bends counted over *all* edges is at most  $b$ , and every edge itself has at most  $\beta(e)$  bends,  $\delta$  is called a *b-bend  $\beta$ -restricted RAC drawing* of  $G$ . We note that

- 0-bend RAC drawings are straight-line RAC drawings (for any choice of  $\beta$ ),
- $m$  and  $2m$ -bend drawings with  $\beta(e) = 1$  or  $\beta(e) = 2$  for each edge  $e$  gives the usual notion of 1-bend and 2-bend RAC drawings, respectively, and
- similarly,  $3m$ -bend drawings with  $\beta(e) = 3$  for each edge  $e$  gives rise to the notion of 3-bend RAC drawings, which exist for every graph [20].

Based on the above, we can now formally define our problem of interest:

**BEND-RESTRICTED RAC DRAWING (BRAC)**

**Input:** A graph  $G$ , an integer  $b \geq 0$ , and an edge-labelling  $\beta : E \mapsto \{0, 1, 2, 3\}$ .

**Question:** Does  $G$  admit a  $b$ -bend  $\beta$ -restricted RAC drawing?

It has been shown that *b-BEND  $\beta$ -RESTRICTED RAC DRAWING* is  $\exists\mathbb{R}$ -complete [44,11] even when restricted to the case where  $b = 0$ . Without loss of generality, we will assume that the input graph  $G$  is connected. We remark that while BRAC is defined as a decision problem, every algorithm provided in this paper is constructive and can output a drawing as a witness for a yes-instance.

**Parameterized Algorithms.** We will not need a lot of the machinery of parameterized algorithms to state our results. However, as it will turn out, our tractability results all come under the guise of so-called *kernelization*, which requires some context.

A *parameterized problem* is an ordinary decision problem, where each instance  $I$  is additionally endowed with a *parameter*  $k$ . Given such a parameterized problem  $\Pi$ , we then say that a problem is *fixed-parameter tractable* (FPT) if there is an algorithm that, upon the input of an instance  $(I, k)$  of  $\Pi$ , decides whether or not  $(I, k)$  is a yes-instance in time  $f(k) \cdot n^{\mathcal{O}(1)}$ , where  $f$  is any computable function, and  $n = |I|$  is the encoding length  $|I|$  of the (parameter-free) instance  $I$ . This should be contrasted with parameterized problems that require time, say,  $n^k$  to solve, which are not fixed-parameter tractable.

For instance, we may ask if a graph has a vertex-cover of size at most  $k$ , and declare  $k$  the parameter of the instance. In this case, the problem is solvable in time  $2^k \cdot n^{\mathcal{O}(1)}$ , and hence FPT; in contrast, asking for a dominating set of size  $k$  (under some complexity assumptions) requires time  $n^k$  for every  $k$ . Closer to the problems treated in this paper are structural parameterizations in the

following sense: Suppose we are given a graph  $G$  and a number  $k$  such that  $G$  has a vertex-cover of size at most  $k$ . Can we leverage this information to solve some (other) graph problem at hand? In this case, we say that we parameterize the problem *by the vertex cover number*.

When using such parameterizations in our results, we will crucially rely on the following notion: A *kernelization* (or kernel, for short) of  $\Pi$  is a polynomial-time algorithm (in  $n$ , and we may assume  $k \leq n$  holds) that takes an instance  $(I, k)$  as input, and produces as output another instance  $(I', k')$  with the following properties: there is some computable function  $g$  such that both  $k'$  and  $|I'|$  are bounded from above by  $g(k)$ , and  $(I, k)$  is a yes-instance of  $\Pi$  if and only if  $(I', k')$  is.

That is, a kernelization algorithm preprocesses instances of arbitrary size into instances that are “parameter-sized,” and in particular (assuming  $\Pi$  was decidable), this implies an algorithm running in time  $n^{\mathcal{O}(1)} + h(g(k))$  for some function  $h$  (where  $h(g(k))$  is the running time of any algorithm solving instances of  $\Pi$  of size  $g(k)$ ). This means in particular that  $\Pi$  is fixed-parameter tractable (and, as a standard result in parameterized algorithms, the converse of this claim holds as well). We refer to the standard textbooks [23,15] for a general treatment of parameterized algorithms.

The *feedback edge number* of a graph  $G$ , denoted  $\mathbf{fen}(G)$ , is the size of a minimum edge set  $F$  such that  $G - F$  is acyclic. It is well-known that such a set  $F$  (and hence also the feedback edge number) can be computed in linear time, since  $G - F$  is a spanning tree of  $G$ . The *vertex cover number* of  $G$ , denoted  $\mathbf{vcn}(G)$ , is the size of a minimum vertex cover of  $G$ , i.e., of a minimum set  $X$  such that  $G - X$  is edgeless. Such a minimum set  $X$  can be computed in time  $\mathcal{O}(1.2738^{|X|} + |X| \cdot |V(G)|)$  [13], and a vertex cover of size at most  $2|X|$  can be computed in linear time by a trivial approximation algorithm. The third structural parameter considered here is the *neighborhood diversity*  $\mathbf{nd}(G)$  of  $G$ , which is the minimum size of a partition  $\mathcal{P}$  of  $V(G)$  such that for each  $a, b$  in the same part of  $\mathcal{P}$  it holds that  $N(a) \setminus \{b\} = N(b) \setminus \{a\}$ . It is well known that each part in such a partition  $\mathcal{P}$  must be either a clique or an independent set, and such a minimum partition can be computed in polynomial time [40].

### 3 An Explicit Algorithm for BRAC

As already pointed out above, our results for fixed-parameter tractability come as kernels. While there is a generic formal equivalence between the existence of a kernel and a decidable problem being fixed-parameter tractable, this doesn't by itself yield explicit bounds on the running time of the algorithm that results from this generic strategy. In order to derive concrete upper bounds on the running time of our algorithms, we provide an algorithm that solves *b-bend  $\beta$ -restricted RAC drawing* with a specific running time bound. We do so via a combination of branching and an encoding in the existential theory of the reals.

**Theorem 1.** *An instance  $(G, b, \beta)$  of BRAC can be solved in time  $m^{\mathcal{O}(m^2)}$ , where  $m$  is the number of edges of  $G$ .*

*Proof.* Observe that, without loss of generality, we may assume that  $b \leq 3m$ . We begin by a branching step in which we exhaustively consider all possible allocations of the bends to edges, resulting in a total number of at most  $4^m$  branches (some of which will be discarded due to exceeding the bound  $b$  or violating  $\beta$ ). In each branch, we alter the graph  $G$  by subdividing each edge precisely the number of times it is assumed to be bent in that branch. At this point, it remains to decide whether this new graph  $G'$  admits a straight-line RAC drawing, where  $G'$  has  $\mathcal{O}(m)$  edges and vertices, and we denote these  $m'$  and  $n'$ , respectively.

To do this, one can construct a sentence in the existential theory of the reals that is true if and only if  $G'$  admits such a drawing. The variables of the sentence will consist of  $n'$  variable pairs  $(x_{v_1}, y_{v_1}), \dots, (x_{v_{n'}}, y_{v_{n'}})$ , encoding the coordinates of the drawing of the vertices in  $\mathbb{R}^2$ . Furthermore, for every pair of edges with endpoints  $u, v$  and  $u', v'$ , we can formulate a condition  $\sigma(u, v, u', v') \Rightarrow \tau(u, v, u', v')$ , where  $\sigma$  is a polynomial condition in  $x_u, x_v, x'_u, x'_v$  encoding whether the straight-line segments corresponding to  $uv$  and  $u'v'$  intersect, and  $\tau$  is a polynomial condition in  $x_u, x_v, x'_u, x'_v$  encoding whether these straight-line segments are perpendicular. Indeed, the former requires an addition of another  $m'^2$  auxiliary variables in the worst case, but both conditions can be expressed by polynomials of degree two. This encoding is described in full detail by Bieker [11].

To conclude the proof, we note that an existential sentence over the reals in  $N$  variables over  $M$  polynomials of maximal degree  $D$  can be decided in time  $(M \cdot D)^{\mathcal{O}(N)}$  (see, e.g., [7, Theorem 13.13]). Note that, within essentially the same running time bound, one can also construct a representation of a solution for this system [7, Theorem 13.11].  $\square$

## 4 A Fixed-Parameter Algorithm via $\text{fen}(G)$

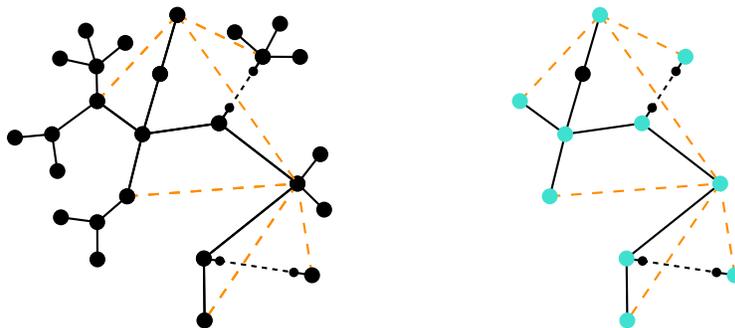
We begin our investigation by establishing a kernel for BEND-RESTRICTED RAC DRAWING when parameterized by the feedback edge number. Our kernel is based on the exhaustive application of two reduction rules.

Let us assume we are given an instance  $(G, b, \beta)$  of BRAC and that we have already computed a minimum feedback edge set  $F$  of  $G$  in linear time. The first reduction rule is trivial: we simply observe that vertices of degree one can always be safely removed since they never hinder the existence of a RAC drawing.

**Observation 2.** *Let  $v \in V(G)$  be a vertex with degree one.  $G - \{v\}$  admits a  $b$ -bend  $\beta$ -restricted RAC drawing if and only if  $G$  does as well.*

*Proof.* Clearly, if  $G$  admits a  $b$ -bend  $\beta$ -restricted RAC drawing, then  $G - \{v\}$  does as well (one may simply remove  $v$  and its incident edge from the drawing). On the other hand, if  $G - \{v\}$  admits a  $b$ -bend  $\beta$ -restricted RAC drawing then we can extend this drawing to one for  $G$  by placing  $v$  sufficiently close to its only neighbor in a way which does not induce any additional crossings.  $\square$

Iteratively applying the reduction rule provided by Observation 2 results in a graph of the form  $G' = (V', E' \cup F)$ , where  $T := (V', E')$  is a tree with at most



(a) Before removing degree-one vertices. (b) After removing degree-one vertices.

Fig. 2: Reduction rule one: Degree-one vertices can be pruned. Orange lines represent feedback edges, dashed black lines represent long paths and special vertices are marked in turquoise.

$2 \cdot \mathbf{fen}(G)$  leaves and where each leaf of  $T$  is incident to at least one edge in  $F$ . We mark a vertex in  $T$  as *special* if it is an endpoint of an edge in  $F$  or if it has degree at least 3 in  $T$  (see Figure 2 for an illustration). Note that the total number of special vertices is upper-bounded by  $4 \cdot \mathbf{fen}(G)$ : the total number of endpoints of edges in  $F$  is bounded by  $2 \cdot \mathbf{fen}(G)$ , and since this also upper-bounds the number of leaves this implies that there can be at most  $2 \cdot \mathbf{fen}(G)$  vertices of degree at least 3 in  $T$ .

In order to define the crucial second reduction rule, we will partition the edges of  $T$  into edge-disjoint paths such that each special vertex can only appear as an endpoint in such paths.

**Definition 3.** We define the path partition of  $T$  in  $G'$  as the unique partition  $P_1 \dot{\cup} \dots \dot{\cup} P_\ell = E'$  such that all  $P_i$  are pairwise edge-disjoint paths in  $T$  whose endpoints are both special vertices, but with no special vertices in their interior. We call  $\ell$  the size of the path partition.

An illustration is provided in Figure 3. Given the established bound on the number of special vertices, the size of the path partition is bounded by  $4 \cdot \mathbf{fen}(G)$ .

At this point, let us assume that we have a path partition  $P_1 \dot{\cup} P_2 \dot{\cup} \dots \dot{\cup} P_\ell$  of  $T$  in  $G'$ , where we index the paths in increasing order of length. Our next task is to divide these paths into short and long paths by identifying whether there exists a large gap in the lengths of these paths.

**Definition 4.** Define  $p_i := |P_i|$  for  $i = 1, \dots, \ell$ , and moreover define  $P_0 := F$  and  $p_0 := |F|$ . Let  $i_0$  be the minimal  $i = 1, \dots, \ell$  such that  $p_i > 9\ell \cdot p_{i-1}$ , if one such  $i$  exists, otherwise we set  $i_0 := \ell$ . We call all paths  $P_i$  with  $1 \leq i \leq i_0$  short and all other paths long. Then we define the subgraph  $G_{\text{short}}$  as the edge-induced subgraph of  $\bigcup_{i=0}^{i_0} P_i$  of  $G'$  (i.e.,  $G_{\text{short}}$  arises by removing all long paths from  $G'$ ).

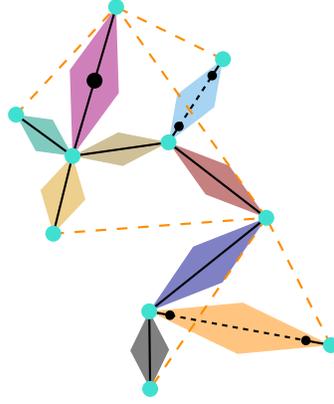


Fig. 3: Path partition of  $T$  with feedback edges in orange.

Our aim is now to argue that if  $\delta_{\text{short}}$  is a RAC drawing of  $G_{\text{short}}$ , then we can always extend  $\delta_{\text{short}}$  to a RAC drawing of  $G'$ . Without loss of generality we assume that all vertices in  $V(G')$  have already been drawn in  $\delta_{\text{short}}$ . First we create an intermediate drawing  $\delta'$  of  $G'$ , which will in general not be a RAC drawing. We define  $\delta'$  as an extension of  $\delta_{\text{short}}$ , where each long path  $P$  with endpoints  $s$  and  $t$  is represented as a simple straight-line segment from  $\delta_{\text{short}}(s)$  to  $\delta_{\text{short}}(t)$  with all interior vertices distributed arbitrarily along that line segment. Doing this will in general violate the RAC property of  $\delta'$ , hence in the next step we need to alter this straight-line segment in order to ensure that the drawing of  $P$  crosses only at right angles. For this we observe that any vertex on  $P$  can be moved to effectively act as a bend in a polyline drawing of  $P$ . We show that these “additional bends” can be used to turn all crossings into right-angle crossings.

**Lemma 5.** *Let  $P$  be a long path with endpoints  $s$  and  $t$  and consider its straight-line representation  $L$  in  $\delta'$ . Assume  $L$  intersects  $k$  straight-line segments in  $\delta'$ . Then, there exists a polyline segment  $L^*$  from  $\delta'(s)$  to  $\delta'(t)$  with at most  $3k$  bends that intersects precisely the line segments intersected by  $L$ , where each such segment is crossed precisely once and at a right angle.*

*Proof.* For the purposes of this proof, it will be useful to treat each bend as an auxiliary vertex in  $\delta'$  and treat straight-line segments as edges. Let  $e \in E(G_{\text{short}})$  be an edge such that  $\delta'(e)$  is crossed by  $L$  at the point  $x$ . We now distinguish the following cases of how  $L$  intersects the straight-line segments in  $\delta'$ , and deal with each case separately:

1.  $x \in \text{int}(\delta'(e))$  and no other edge crosses through  $x$ .
2.  $\exists f \in E(G_{\text{short}}) : f \neq e \wedge x \in \text{int}(\delta'(e)) \cap \text{int}(\delta'(f))$ .
3.  $\exists v \in V(G_{\text{short}}) : x = \delta'(v)$ .

We show how to deal with each of these three cases below:

1. Let  $\varepsilon > 0$  such that  $B_\varepsilon(x) = \{y \in \mathbb{R}^2 : |x - y| < \varepsilon\}$  contains no vertices and intersects no other edges outside of  $L$  apart from  $e$ . We convert the intersection at  $x$  to a right-angle crossing by introducing three bends on the boundary  $\partial B_\varepsilon(x)$  as illustrated in Figure 4: Put two vertices  $v_1, v_3$  on the intersection of  $L$  with  $\partial B_\varepsilon(x)$  to maintain the position of  $L$  outside of  $B_\varepsilon(x)$ . Construct the middle vertex  $v_2$  by taking the intersection of  $\partial B_\varepsilon(x)$  with the normal line to  $e$  going through  $v_3$ . Therefore we obtain our new polyline  $L^*$  by joining the parts of  $L$  outside the  $\varepsilon$ -neighborhood with the polyline connecting the three vertices on the boundary of the  $\varepsilon$ -neighborhood.

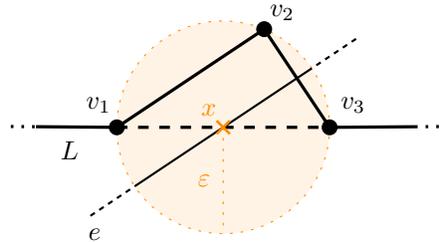


Fig. 4: Dealing with a single crossing.

Since we chose  $\varepsilon$  such that there are no further intersections within this  $\varepsilon$ -neighborhood and the polyline remains unchanged outside of this neighborhood, we are guaranteed to not introduce any new crossings nor alter existing ones.

2. Generalizing the first case, we now consider an intersection point  $x$  with multiple edges crossing it. We denote with  $C = \{e_1, \dots, e_t\}$  the set of edges intersecting at  $x$ . Let  $\varepsilon > 0$  such that  $B_\varepsilon(x)$  contains no vertices and intersects no other edges apart from the edges in  $C$ . Again, two entry and exit nodes  $v_1, v_{t+1}$  are added on the boundary of  $B_\varepsilon(x)$  to preserve the original position of  $L$  outside of  $B_\varepsilon(x)$ . Assume the edges in  $C$  are ordered clockwise with  $e_1$  being the closest edge to  $v_1$ . Now we iteratively construct the next vertex  $v_{i+1}$  by taking the intersection of the angular bisector between  $e_i$  and  $e_{i+1}$  with the normal line to  $e_i$  going through  $v_i$ . If this point happens to lie outside of  $B_\varepsilon(x)$  we take the intersection of the normal line with  $\partial B_\varepsilon(x)$  instead. We refer to Figure 5 for an illustration.  
In total, we need at most  $t + 1 \leq 3t$  vertices.
3. In the third case, the straight-line segment  $L$  intersects with a vertex  $v$  at  $x = \delta'(v)$ . If  $L \cap \delta'(e) = \delta'(v)$ , i.e.  $L$  does not run parallel to  $\delta'(e)$ , we can simply take care of this equivalently as if  $L$  would cross  $\delta'(e)$  in the interior. If otherwise  $L \cap \delta'(e) \supsetneq \delta'(v)$ , we observe that  $L$  must necessarily contain both endpoints of  $\delta'(e)$ , since the endpoints  $\delta'(s)$  and  $\delta'(t)$  of  $L$  cannot lie in the interior of  $\delta'(e)$ . At the first of these endpoints encountered by  $L$ , say  $w$ , we again proceed analogously to the second case (i.e., as if  $w$  was a crossing point), however instead of exiting the circle surrounding  $w$  directly opposite

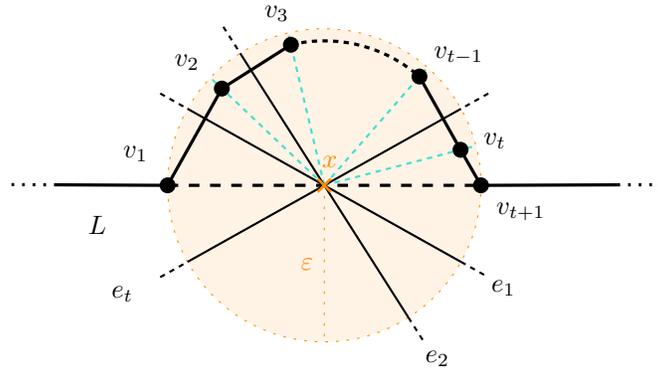


Fig. 5: Crossing multiple edges at once.

to the entry point we exit at a distance of  $\varepsilon'$  from there (for a sufficiently small  $\varepsilon'$ ) and from there draw  $L$  parallel to  $e$ .

Since these three cases are exhaustive, the lemma follows.  $\square$

**Lemma 6.** *Each long path intersects at most  $3\ell \cdot p_{i_0}$  straight-line edge segments in  $\delta'$ .*

*Proof.* Since each long path is represented as a straight-line segment in  $\delta'$ , it can cross every other long path at most once. Since every edge  $e \in E(G_{\text{short}})$  can be bent at most  $\beta(e) \leq 3$  times,  $L$  can cross  $e$  at most three times. Let  $t$  be the number of long paths, then  $L$  intersects no more than

$$t + 3 \sum_{i=0}^{i_0} p_i \leq t + 3(\ell - t) \cdot p_{i_0} \leq 3\ell \cdot p_{i_0}$$

straight-line edge segments.  $\square$

**Theorem 7.**  *$b$ -BEND  $\beta$ -RESTRICTED RAC DRAWING admits a kernel of size at most  $(36 \cdot \mathbf{fen}(G))^{4 \cdot \mathbf{fen}(G)}$ . The kernel can be constructed in linear time.*

*Proof.* Consider an input  $(G = (V, E), b, \beta)$  with feedback edge set  $F$ . In the first step according to Observation 2 we iteratively prune all vertices of degree one and obtain the reduced graph  $G' = (V', E' \cup F)$ . Next, we construct a path partition  $\mathcal{P} = (P_1, \dots, P_\ell)$  of the tree  $T = (V', E')$  of size at most  $4 \cdot \mathbf{fen}(G)$ . Then we split the paths in  $\mathcal{P}$  into short and long paths. We define the subgraph  $G_{\text{short}}$  as the graph obtained by removing all long paths from  $G'$  and show that it is a kernel.

For that consider a RAC drawing  $\delta_{\text{short}}$  of  $G_{\text{short}}$ . We show that we can construct a RAC drawing  $\delta$  of  $G'$  that extends  $\delta_{\text{short}}$  (see Figure 6b). To achieve this, we first define the intermediate drawing  $\delta'$ , which extends  $\delta_{\text{short}}$  by simply drawing all long paths as straight-line segments. To obtain  $\delta$  from  $\delta'$  we now

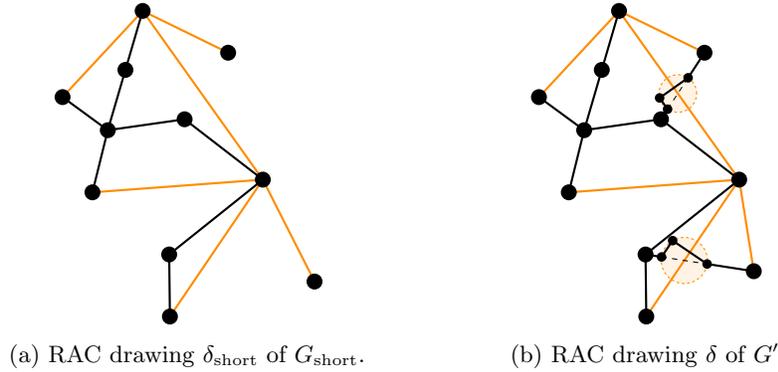


Fig. 6: Extending the RAC drawing.

iteratively replace the straight-line representations by the polyline constructions described in Lemma 5. Let  $P_{i_0+1}$  be the first long path with straight-line drawing  $L_{i_0+1}$ . According to Lemma 6 it is involved in at most  $3\ell \cdot p_{i_0}$  crossings in  $\delta'$ . By the definition of long paths we have  $|P_{i_0+1}| > 9\ell \cdot p_{i_0}$  and thus we have enough vertices per crossing to construct  $L_{i_0+1}^*$  via Lemma 5. Define  $\delta_{i_0+1}$  by replacing  $L_{i_0+1}$  with  $L_{i_0+1}^*$  in  $\delta'$ . Since  $L_{i_0+1}^*$  does not introduce any new crossings, we can repeat this process for all further long paths until we obtain  $\delta := \delta_\ell$  as our final RAC drawing of  $T$ . Thus, we obtain  $G_{\text{short}}$  as a kernel of our original instance with  $b$  and  $\beta$  unchanged, since we did not use any additional bends in the construction of  $\delta$ .

Next, we show that  $G_{\text{short}}$  can be constructed in linear time. We already observed that a Feedback Edge Set can be constructed in linear time. Pruning vertices of degree one can be done in linear time as well, while the task of finding a path partition of  $T$  can be achieved by depth-first search in linear time as well. Finally, the size of  $G_{\text{short}}$  can be bounded by

$$\begin{aligned} \sum_{i=0}^{i_0} p_i &\leq \sum_{i=0}^{\ell} p_0 (9\ell \cdot \mathbf{fen}(G))^i \\ &\leq 2 \cdot \mathbf{fen}(G) (9\ell \cdot \mathbf{fen}(G))^\ell \\ &\leq 2(36 \cdot \mathbf{fen}(G))^{4 \cdot \mathbf{fen}(G)}. \end{aligned} \quad \square$$

Using Theorem 7, the runtime guarantee given by Theorem 1 and the fact that a feedback edge set of size  $\mathbf{fen}(G)$  can be computed in linear time, we obtain:

**Corollary 8.**  *$b$ -BEND  $\beta$ -RESTRICTED RAC DRAWING is fixed-parameter tractable parameterized by  $\mathbf{fen}(G)$ , and in particular can be solved in time  $2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} + \mathcal{O}(|V(G)|)$ .*

*Proof.* After constructing the kernel in  $\mathcal{O}(|V(G)|)$  time, we apply the generic branching algorithm to the kernel with a runtime of

$$\begin{aligned} ((36 \cdot \mathbf{fen}(G))^{4 \cdot \mathbf{fen}(G)})^{\mathcal{O}((36 \cdot \mathbf{fen}(G))^{8 \cdot \mathbf{fen}(G)})} &= (36 \cdot \mathbf{fen}(G))^{\mathcal{O}((36 \cdot \mathbf{fen}(G))^{8 \cdot \mathbf{fen}(G)+1})} \\ &\leq 2^{(\log \mathbf{fen}(G)) \cdot \mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}} \\ &= 2^{\mathbf{fen}(G)^{\mathcal{O}(\mathbf{fen}(G))}}, \end{aligned}$$

which concludes the proof.  $\square$

## 5 Fixed-Parameter Tractability via $\mathbf{vcn}(G)$

As in Section 4, the core tool used to establish fixed-parameter tractability for this parameterization is a kernelization procedure, although the ideas and reduction rules used here are very different. Let us assume we are given an instance  $(G, b, \beta)$  of BRAC; as our first step, we compute a vertex cover  $C$  of size  $k \leq 2 \cdot \mathbf{vcn}(G)$  using the standard approximation algorithm.

We now partition the vertices of our instance  $G$  outside of the vertex cover  $C$  into *types*, as follows. Two vertices in  $G \setminus C$  are of the same type if they have the same set of neighbors in  $C$ ; observe that the property of “being in the same type” is an equivalence relation, and when convenient we also use the term *type* to refer to the equivalence classes of this relation. To avoid any confusion, we explicitly remark that two vertices may have the same type even when their incident edges are assigned different values by  $\beta$ . The number of types is upper-bounded by  $2^k$ .

We distinguish types by the number of neighbors in  $C$ ; an illustration is provided in Figure 7. Let a *member* of a type  $T$  be defined as a vertex in  $T$  as well as its incident edges. By an exhaustive application of the first reduction rule introduced in Section 4 (cf. Observation 2), we may assume that there is no type with less than 2 neighbors in  $C$ .

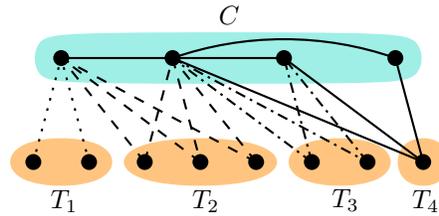


Fig. 7: A graph split into its vertex cover  $C$  (in turquoise) and its different types  $T_1, \dots, T_4$  (in orange).

Turning to types with at least 3 neighbors in  $C$ , we provide a bound on the size of each such type in a yes-instance of BRAC.

**Lemma 9.** *If  $(G, b, \beta)$  is a yes-instance of BRAC, then each type  $T$  with  $i \geq 3$  neighbors in  $C$  has at most  $\max(2, 7 - i) + b$  members.*

*Proof.* Didimo, Eades and Liotta showed that no complete bipartite graph  $K_{c,d}$  with  $c + d > 7$  and  $\min(c, d) > 2$  admits a straight-line RAC drawing [19]. Hence, if vertices in  $T$  have 3 neighbors in  $C$  then a  $b$ -bend  $\beta$ -restricted RAC drawing of  $G$  can contain at most 4 members of  $T$  without bends; otherwise, the drawing of 5 members of  $T$  and their 3 neighbors in  $C$  would contradict the first sentence. Similarly, if vertices in  $T$  have at least 5 neighbors in  $C$  then a  $b$ -bend  $\beta$ -restricted RAC drawing of  $G$  cannot contain 3 members of  $T$  without bends.  $\square$

Lemma 9 implies that we can immediately reject instances containing types with more than 3 neighbors whose cardinality is greater than  $4 + b$  (or, for the purposes of kernelization, one may replace these with trivial no-instances). Hence, it now remains to deal with types with precisely two neighbors in  $C$ .

We say that two edges  $uv$  and  $uw'$  form a *fan anchored* at  $u$ . It is easy to observe that if an edge  $e$  crosses both  $uv$  and  $uw'$  in a  $b$ -bend  $\beta$ -restricted RAC drawing, then at least one of these three edges must have a bend [3].

**Lemma 10.** *Consider a  $b$ -bend  $\beta$ -restricted RAC drawing  $\delta$  of  $G$ , and let  $T$  be a type containing vertices with precisely two neighbors in  $C$ . Let  $T'$  be the subset of  $T$  containing all members of  $T$  which do not have bends in  $\delta$ . Then  $T'$  contains at most four members involved in crossings with other members of  $T'$  in  $\delta$ .*

*Proof.* Let  $u$  and  $v$  be the neighbors of  $T'$  in the vertex cover  $C$ . Let us consider the vertices lying on one side of the half plane induced by the line  $\overleftrightarrow{\delta(u)\delta(v)}$  going through  $u$  and  $v$ . According to Thales's theorem, every right-angle crossing formed by two edges originating in  $u$  and  $v$  respectively, has to lie on the semicircle with diameter  $\overline{\delta(u)\delta(v)}$ . Suppose the edges  $(u, x)$  and  $(v, w)$  cross at a right angle. Then there cannot be another edge incident to  $u$  which crosses the semicircle to the right of the first crossing.

Indeed, if there was such an edge  $(u, y)$ , then  $(v, w)$  would cross the fan formed by  $(u, x)$  and  $(u, y)$  as shown in Figure 8. Analogously, there also cannot be an edge incident to  $v$ , which crosses the semicircle left to the first crossing. Hence, there can be at most one right-angle crossing between members of  $T'$  below and above  $\overleftrightarrow{\delta(u)\delta(v)}$ , respectively.  $\square$

Next, we use the above statement to obtain a bound on the total number of crossings that such a type  $T$  can be involved in. We do so by showing that the members of  $T$  which themselves do not have bends are only involved in a bounded number of crossings.

**Lemma 11.** *Consider a  $b$ -bend  $\beta$ -restricted RAC drawing  $\delta$  of  $G$ , and let  $T$  be a type containing vertices with precisely two neighbors in  $C$ . Then at most  $3k + 6 + b$  members of  $T$  can be involved in a crossing in  $\delta$ .*

*Proof.* Let  $T'$  be the subset of  $T$  containing all members of  $T$  which do not have bends in  $\delta$ , and let  $\gamma = |T| - |T'|$ . Further, let  $T_0$  be the set of members of  $T'$



member that is not involved in any crossings. The next lemma highlights why this is useful in the context of our kernelization.

**Lemma 12.** *Let  $T$  be a type with two neighbors in  $C$  and assume that  $G$  admits a  $b$ -bend  $\beta$ -restricted RAC drawing  $\delta$ . If there is a member in  $T$  whose edges are drawn without crossings in  $\delta$ , then the graph obtained from  $G$  by adding a vertex  $w'$  to  $T$  admits a  $b$ -bend  $\beta$ -restricted RAC drawing as well.*

*Proof.* Let  $u, v$  be the neighbors of  $T$  in the vertex cover and let  $w \in T$  be the member without crossings. We can draw  $w'$  infinitesimally close to  $w$  such that the emerging layering triangles are drawn without crossings (see Figure 10).  $\square$

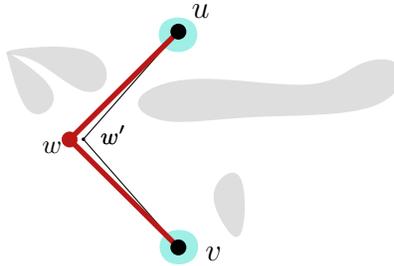


Fig. 10: Illustration for the proof of Lemma 12.

At this point, we have all the ingredients for the main result of this section:

**Theorem 13.**  *$b$ -BEND  $\beta$ -RESTRICTED RAC DRAWING admits a kernel of size  $\mathcal{O}(b \cdot 2^k)$ , where  $k$  is the size of a provided vertex cover of the input graph.*

*Proof.* Consider an input  $(G, b, \beta)$  and let  $C$  be the provided vertex cover of  $G$ . We apply the simple reduction rule of deleting vertices of degree 1 from  $G$ , resulting in an instance where each type has either 2 or at least 3 neighbors in  $C$ . For each type of the latter kind, we check if it contains more members than  $\max(3, 7-i)+b$ ; if yes, we reject (or, equivalently, replace the instance with a trivial constant-size no-instance), and this is correct by Lemma 9. Moreover, for each type  $T$  with precisely 2 neighbors in  $C$  containing more than  $3k+6+b+1$  many members, we delete members from  $T$  until its size is precisely  $3k+6+b+1$ —the correctness of this step follows from Lemma 11 and 12.

In the resulting graph, each of the at most  $2^k$  many types with at least 3 neighbors in  $C$  has size at most  $b+4$ , while each of the at most  $k^2$  types with precisely 2 neighbors has size at most  $3k+6+b+1$ . The kernel bound follows.  $\square$

From Theorem 13, the runtime bound given by Theorem 1 and the fact that a vertex cover of size at most  $2 \cdot \mathbf{vcn}(G)$  can be obtained in linear time, we obtain:

**Corollary 14.**  *$b$ -BEND  $\beta$ -RESTRICTED RAC DRAWING is fixed-parameter tractable parameterized by  $b + \mathbf{vcn}(G)$ , and in particular can be solved in time  $2^{2^{\mathcal{O}(\mathbf{vcn}(G) + \log b)}} + \mathcal{O}(|V(G)|)$ .*

*Proof.* Applying the runtime result of  $m^{\mathcal{O}(m^2)}$  given in Theorem 1 to the given kernel yields a final runtime of

$$\begin{aligned} \mathcal{O}(b \cdot 2^{\mathbf{vcn}(G)})^{\mathcal{O}(b^2 4^{\mathbf{vcn}(G)})} &= 2^{(\log b + \mathbf{vcn}(G)) \cdot \mathcal{O}(b^2 4^{\mathbf{vcn}(G)})} \\ &\leq 2^{b^2 \log b \cdot 2^{\mathcal{O}(\mathbf{vcn}(G))}} \\ &= 2^{2^{(\log \log b) + 2 \log b} \cdot 2^{\mathcal{O}(\mathbf{vcn}(G))}} \\ &= 2^{2^{\mathcal{O}(\log b + \mathbf{vcn}(G))}}, \end{aligned}$$

which concludes the proof.  $\square$

## 6 An Extension to Neighborhood Diversity

We extend the approach used for the vertex cover number to establish fixed-parameter tractability with respect to *neighborhood diversity*. Briefly recalling the definition of neighborhood diversity, let two vertices  $v, v'$  be of the same type if  $N(v) \setminus \{v'\} = N(v') \setminus \{v\}$ .

**Definition 15** ([40,39]). *The neighborhood diversity  $\mathbf{nd}(G)$  of a graph  $G$  is the minimum number  $k$ , such that there exists a partition into  $k$  sets, where all vertices in each set have the same type.*

By the definition of neighborhood diversity, each set in the witnessing partition is either an independent set or clique in  $G$ . Edges can occur either on all vertices between two sets or on none (see Figure 11). In general, a graph  $G$  with neighborhood diversity  $\mathbf{nd}(G)$  has a bounded vertex cover number  $\mathbf{vcn}(G)$ . Thus, Theorem 13 would already imply tractability of  $b$ -bend RAC drawings under a bounded neighborhood diversity. However,  $\mathbf{vcn}(G)$  might be exponentially larger [40]. For  $b$ -bend RAC drawable graphs, we can show a better, linear bound on  $\mathbf{vcn}(G)$ .

**Lemma 16.** *Let  $G$  be a  $b$ -bend RAC drawable graph with a neighborhood diversity  $\mathbf{nd}(G)$ . Then  $\mathbf{vcn}(G) \leq 5 \cdot \mathbf{nd}(G) + b$ .*

*Proof.* We begin by showing a linear bound on  $\mathbf{vcn}(G)$  for  $b = 0$ . Let  $S_1, \dots, S_{\mathbf{nd}(G)}$  be a partition witnessing the neighborhood diversity number  $\mathbf{nd}(G)$ . We build a vertex cover  $C$  as follows. The size of each set  $S_i$  forming a clique in  $G$  is bounded by 5, as a  $K_6$  is not straight-line RAC drawable [20]. Put all vertices of such an  $S_i$  in  $C$ . Let  $S_i, S_j$  be a pair of two sets, which are both forming an independent set in  $G$ , and have edges between each other. If there is an edge between a vertex in  $S_i$  and a vertex in  $S_j$ , there is an edge between all vertices of  $S_i$  and  $S_j$ . Let  $|S_i| \leq |S_j|$ . Recalling that no complete bipartite graph  $K_{a,b}$  with  $a + b > 7$  and

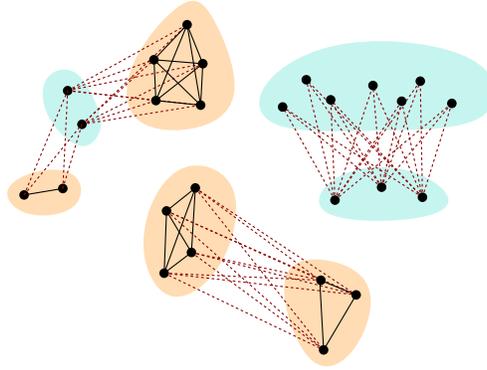


Fig. 11: Overview of a graph partitioned into its neighborhood diversity sets. Orange sets build cliques, turquoise sets are independent sets in  $G$ . Each set may be connected to one or more other sets.

$\min(a, b) > 2$  admits a straight-line RAC drawing [19],  $S_i \leq 3$ . Put  $S_i$  into  $C$  to cover both sets. In total, we put at most  $5 \cdot \mathbf{nd}(G)$  vertices into  $C$ .

For arbitrary number of bends  $b$ , the total number of vertices in clique sets might increase by at most  $b$  without making  $G$  not  $b$ -bend RAC drawable. Similarly, the number of vertices in the smaller set  $S_i$  of a connected set pair  $S_i, S_j$ , might increase by at most  $b$  over all such sets. So in total,  $\mathbf{vcn}(G) \leq 5 \cdot \mathbf{nd}(G) + b$ .  $\square$

From Theorem 13 and Lemma 16 the following theorem follows directly:

**Theorem 17.**  *$b$ -BEND  $\beta$ -RESTRICTED RAC DRAWING admits a kernel of size  $\mathcal{O}(b^2 \cdot \mathbf{nd}(G) \cdot 2^{\mathbf{nd}(G)})$ .*

**Corollary 18.**  *$b$ -BEND  $\beta$ -RESTRICTED RAC DRAWING is fixed-parameter tractable parameterized by  $\mathbf{nd}(G) + b$ , and in particular can be solved in time  $2^{b^{\mathcal{O}(\mathbf{nd}(G))}} + \mathcal{O}(|V(G)| + \mathbf{nd}(G))$ .*

## 7 Concluding Remarks

We have established the fixed-parameter tractability of  $b$ -BEND  $\beta$ -RESTRICTED RAC DRAWING when parameterized by the feedback edge number  $\mathbf{fen}(G)$ , or by the vertex cover number  $\mathbf{vcn}(G)$  plus an upper bound  $b$  on the total number of bends. We have also shown that the latter result implies the fixed-parameter tractability of the problem w.r.t. the neighborhood diversity  $\mathbf{nd}(G)$  plus  $b$ .

A next step in the computational study of RAC Drawings would be to consider whether the problem is fixed-parameter tractable w.r.t.  $\mathbf{vcn}(G)$  alone. Interestingly, a reduction rule for degree-2 vertices without a bound on  $b$  is the main obstacle towards obtaining such a fixed-parameter algorithm, and dealing with this case seems to be required if one wishes to generalize the result towards

fixed-parameter tractability w.r.t. *treedepth* [42] plus  $b$ . A different question one may ask is whether the fixed-parameter algorithm w.r.t.  $\mathbf{fen}(G)$  can be generalized towards the recently introduced parameter *slim tree-cut width* [31], which can be equivalently seen as a local version of the feedback edge number [12]. A natural long-term goal within this research direction is then to obtain an understanding of the complexity of BRAC w.r.t. treewidth [43]. Last but not least, it would be interesting to see whether our fixed-parameter tractability results can be strengthened by obtaining polynomial kernels for the same parameterizations.

*Acknowledgments* The authors graciously accept support from the WWTF (Project ICT22-029) and the FWF (Project Y1329) science funds.

## References

1. Angelini, P., Bekos, M.A., Förster, H., Kaufmann, M.: On RAC drawings of graphs with one bend per edge. *Theor. Comput. Sci.* **828-829**, 42–54 (2020). <https://doi.org/10.1016/j.tcs.2020.04.018>
2. Angelini, P., Bekos, M.A., Katheder, J., Kaufmann, M., Pfister, M.: RAC drawings of graphs with low degree. In: Szeider, S., Ganian, R., Silva, A. (eds.) 47th International Symposium on Mathematical Foundations of Computer Science, MFCS 2022, August 22–26, 2022, Vienna, Austria. LIPIcs, vol. 241, pp. 11:1–11:15. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). <https://doi.org/10.4230/LIPIcs.MFCS.2022.11>
3. Angelini, P., Cittadini, L., Didimo, W., Frati, F., Di Battista, G., Kaufmann, M., Symvonis, A.: On the perspectives opened by right angle crossing drawings. *J. Graph Algorithms Appl.* **15**(1), 53–78 (2011). <https://doi.org/10.7155/jgaa.00217>
4. Argyriou, E.N., Bekos, M.A., Symvonis, A.: The straight-line RAC drawing problem is np-hard. *J. Graph Algorithms Appl.* **16**(2), 569–597 (2012). <https://doi.org/10.7155/jgaa.00274>
5. Balko, M., Chaplick, S., Ganian, R., Gupta, S., Hoffmann, M., Valtr, P., Wolff, A.: Bounding and computing obstacle numbers of graphs. In: Chechik, S., Navarro, G., Rotenberg, E., Herman, G. (eds.) 30th Annual European Symposium on Algorithms, ESA 2022, September 5–9, 2022, Berlin/Potsdam, Germany. LIPIcs, vol. 244, pp. 11:1–11:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). <https://doi.org/10.4230/LIPIcs.ESA.2022.11>
6. Bannister, M.J., Cabello, S., Eppstein, D.: Parameterized complexity of 1-planarity. *J. Graph Algorithms Appl.* **22**(1), 23–49 (2018). <https://doi.org/10.7155/jgaa.00457>
7. Basu, S., Pollack, R., Roy, M.F.: *Algorithms in Real Algebraic geometry*, Algorithms and Computation in Mathematics, vol. 10. Springer (2006). <https://doi.org/10.1007/3-540-33099-2>, <http://link.springer.com/10.1007/3-540-33099-2>
8. Bekos, M.A., Didimo, W., Liotta, G., Mehrabi, S., Montecchiani, F.: On RAC drawings of 1-planar graphs. *Theor. Comput. Sci.* **689**, 48–57 (2017). <https://doi.org/10.1016/j.tcs.2017.05.039>
9. Bhore, S., Ganian, R., Montecchiani, F., Nöllenburg, M.: Parameterized algorithms for book embedding problems. *J. Graph Algorithms Appl.* **24**(4), 603–620 (2020). <https://doi.org/10.7155/jgaa.00526>

10. Bhore, S., Ganian, R., Montecchiani, F., Nöllenburg, M.: Parameterized algorithms for queue layouts. *J. Graph Algorithms Appl.* **26**(3), 335–352 (2022). <https://doi.org/10.7155/jgaa.00597>
11. Bieker, N.: Complexity of graph drawing problems in relation to the existential theory of the reals. Ph.D. thesis, Bachelor’s thesis, Karlsruhe Institute of Technology (August 2020) (2020)
12. Brand, C., Ceylan, E., Ganian, R., Hatschka, C., Korchemna, V.: Edge-cut width: An algorithmically driven analogue of treewidth based on edge cuts. In: Bekos, M.A., Kaufmann, M. (eds.) *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers. Lecture Notes in Computer Science*, vol. 13453, pp. 98–113. Springer (2022). [https://doi.org/10.1007/978-3-031-15914-5\\_8](https://doi.org/10.1007/978-3-031-15914-5_8)
13. Chen, J., Kanj, I.A., Xia, G.: Improved upper bounds for vertex cover. *Theor. Comput. Sci.* **411**(40-42), 3736–3756 (2010). <https://doi.org/10.1016/j.tcs.2010.06.026>
14. Courcelle, B., Makowsky, J.A., Rotics, U.: Linear time solvable optimization problems on graphs of bounded clique-width. *Theory Comput. Syst.* **33**(2), 125–150 (2000). <https://doi.org/10.1007/s002249910009>
15. Cygan, M., Fomin, F.V., Kowalik, L., Lokshtanov, D., Marx, D., Pilipczuk, M., Pilipczuk, M., Saurabh, S.: *Parameterized Algorithms*. Springer Publishing Company, Incorporated, 1st edn. (2015)
16. Di Giacomo, E., Didimo, W., Eades, P., Liotta, G.: 2-layer right angle crossing drawings. *Algorithmica* **68**(4), 954–997 (2014). <https://doi.org/10.1007/s00453-012-9706-7>
17. Di Giacomo, E., Didimo, W., Grilli, L., Liotta, G., Romeo, S.A.: Heuristics for the maximum 2-layer RAC subgraph problem. *Comput. J.* **58**(5), 1085–1098 (2015). <https://doi.org/10.1093/comjnl/bxu017>
18. Didimo, W.: Right angle crossing drawings of graphs. In: Hong, S., Tokuyama, T. (eds.) *Beyond Planar Graphs, Communications of NII Shonan Meetings*, pp. 149–169. Springer (2020). [https://doi.org/10.1007/978-981-15-6533-5\\_9](https://doi.org/10.1007/978-981-15-6533-5_9)
19. Didimo, W., Eades, P., Liotta, G.: A characterization of complete bipartite rac graphs. *Information Processing Letters* **110**(16), 687–691 (2010). <https://doi.org/10.1016/j.ipl.2010.05.023>
20. Didimo, W., Eades, P., Liotta, G.: Drawing graphs with right angle crossings. *Theoretical Computer Science* **412**(39), 5156–5166 (2011). <https://doi.org/10.1016/j.tcs.2011.05.025>
21. Didimo, W., Liotta, G., Montecchiani, F.: A survey on graph drawing beyond planarity. *ACM Comput. Surv.* **52**(1), 4:1–4:37 (2019). <https://doi.org/10.1145/3301281>
22. Diestel, R.: *Graph Theory, 5th Edition, Graduate Texts in Mathematics*, vol. 173. Springer (2017). <https://doi.org/10.1007/978-3-662-53622-3>
23. Downey, R.G., Fellows, M.R.: *Fundamentals of Parameterized Complexity*. Texts in Computer Science, Springer (2013). <https://doi.org/10.1007/978-1-4471-5559-1>
24. Eiben, E., Ganian, R., Hamm, T., Klute, F., Nöllenburg, M.: Extending nearly complete 1-planar drawings in polynomial time. In: Esparza, J., Král’, D. (eds.) *45th International Symposium on Mathematical Foundations of Computer Science, MFCS 2020, August 24-28, 2020, Prague, Czech Republic. LIPIcs*, vol. 170, pp. 31:1–31:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). <https://doi.org/10.4230/LIPIcs.MFCS.2020.31>

25. Eiben, E., Ganian, R., Hamm, T., Klute, F., Nöllenburg, M.: Extending partial 1-planar drawings. In: Czumaj, A., Dawar, A., Merelli, E. (eds.) 47th International Colloquium on Automata, Languages, and Programming, ICALP 2020, July 8-11, 2020, Saarbrücken, Germany (Virtual Conference). LIPIcs, vol. 168, pp. 43:1–43:19. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). <https://doi.org/10.4230/LIPIcs.ICALP.2020.43>
26. Fleszar, K., Mnich, M., Spoerhase, J.: New algorithms for maximum disjoint paths based on tree-likeness. *Math. Program.* **171**(1-2), 433–461 (2018). <https://doi.org/10.1007/s10107-017-1199-3>
27. Förster, H., Kaufmann, M.: On compact RAC drawings. In: Grandoni, F., Herman, G., Sanders, P. (eds.) 28th Annual European Symposium on Algorithms, ESA 2020, September 7-9, 2020, Pisa, Italy (Virtual Conference). LIPIcs, vol. 173, pp. 53:1–53:21. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2020). <https://doi.org/10.4230/LIPIcs.ESA.2020.53>
28. de Fraysseix, H., Pach, J., Pollack, R.: Small sets supporting fary embeddings of planar graphs. In: Simon, J. (ed.) Proceedings of the 20th Annual ACM Symposium on Theory of Computing, May 2-4, 1988, Chicago, Illinois, USA. pp. 426–433. ACM (1988). <https://doi.org/10.1145/62212.62254>
29. Fáry, I.: On straight lines representation of planar graphs. *Acta Sci. Math. (Szeged)* **11**, 229–233 (1948)
30. Ganian, R.: Using neighborhood diversity to solve hard problems. *CoRR abs/1201.3091* (2012). <https://doi.org/10.48550/arXiv.1201.3091>
31. Ganian, R., Korchemna, V.: Slim tree-cut width. In: Dell, H., Nederlof, J. (eds.) 17th International Symposium on Parameterized and Exact Computation, IPEC 2022, September 7-9, 2022, Potsdam, Germany. LIPIcs, vol. 249, pp. 15:1–15:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik (2022). <https://doi.org/10.4230/LIPIcs.IPEC.2022.15>
32. Ganian, R., Ordyniak, S.: The power of cut-based parameters for computing edge-disjoint paths. *Algorithmica* **83**(2), 726–752 (2021). <https://doi.org/10.1007/s00453-020-00772-w>
33. Garey, M.R., Johnson, D.S.: Crossing number is np-complete. *SIAM Journal on Algebraic Discrete Methods* **4**(3), 312–316 (1983). <https://doi.org/10.1137/0604033>
34. Grohe, M.: Computing crossing numbers in quadratic time. *J. Comput. Syst. Sci.* **68**(2), 285–302 (2004). <https://doi.org/10.1016/j.jcss.2003.07.008>
35. Hliněný, P., Sankaran, A.: Exact crossing number parameterized by vertex cover. In: Archambault, D., Tóth, C.D. (eds.) Graph Drawing and Network Visualization - 27th International Symposium, GD 2019, Prague, Czech Republic, September 17-20, 2019, Proceedings. Lecture Notes in Computer Science, vol. 11904, pp. 307–319. Springer (2019). [https://doi.org/10.1007/978-3-030-35802-0\\_24](https://doi.org/10.1007/978-3-030-35802-0_24)
36. Huang, W.: Using eye tracking to investigate graph layout effects. In: Hong, S., Ma, K. (eds.) APVIS 2007, 6th International Asia-Pacific Symposium on Visualization 2007, Sydney, Australia, 5-7 February 2007. pp. 97–100. IEEE Computer Society (2007). <https://doi.org/10.1109/APVIS.2007.329282>
37. Huang, W., Eades, P., Hong, S.: Larger crossing angles make graphs easier to read. *J. Vis. Lang. Comput.* **25**(4), 452–465 (2014). <https://doi.org/10.1016/j.jvlc.2014.03.001>
38. Huang, W., Hong, S., Eades, P.: Effects of crossing angles. In: IEEE VGTC Pacific Visualization Symposium 2008, PacificVis 2008, Kyoto, Japan, March 5-7, 2008. pp. 41–46. IEEE Computer Society (2008). <https://doi.org/10.1109/PACIFICVIS.2008.4475457>

39. Knop, D., Kouřtecký, M., Masarík, T., Toufar, T.: Simplified algorithmic meta-theorems beyond MSO: treewidth and neighborhood diversity. *Log. Methods Comput. Sci.* **15**(4) (2019). [https://doi.org/10.23638/LMCS-15\(4:12\)2019](https://doi.org/10.23638/LMCS-15(4:12)2019)
40. Lampis, M.: Algorithmic meta-theorems for restrictions of treewidth. In: de Berg, M., Meyer, U. (eds.) *Algorithms - ESA 2010, 18th Annual European Symposium, Liverpool, UK, September 6-8, 2010. Proceedings, Part I. Lecture Notes in Computer Science*, vol. 6346, pp. 549–560. Springer (2010). [https://doi.org/10.1007/978-3-642-15775-2\\_47](https://doi.org/10.1007/978-3-642-15775-2_47)
41. Mutzel, P.: An alternative method to crossing minimization on hierarchical graphs. *SIAM J. Optim.* **11**(4), 1065–1080 (2001). <https://doi.org/10.1137/S1052623498334013>
42. Nešetřil, J., Ossona de Mendez, P.: *Sparsity: Graphs, Structures, and Algorithms, Algorithms and Combinatorics*, vol. 28. Springer Berlin Heidelberg (2015). <https://doi.org/10.1007/978-3-642-27875-4>
43. Robertson, N., Seymour, P.D.: Graph minors. III. planar tree-width. *J. Comb. Theory, Ser. B* **36**(1), 49–64 (1984). [https://doi.org/10.1016/0095-8956\(84\)90013-3](https://doi.org/10.1016/0095-8956(84)90013-3)
44. Schaefer, M.: Rac-drawability is  $\exists\mathbb{R}$ -complete. In: *Graph Drawing and Network Visualization: 29th International Symposium, GD 2021, Tübingen, Germany, September 14–17, 2021, Revised Selected Papers*. p. 72–86. Springer-Verlag, Berlin, Heidelberg (2021). [https://doi.org/10.1007/978-3-030-92931-2\\_{\\_}5](https://doi.org/10.1007/978-3-030-92931-2_{_}5)