# Most Classic Problems Remain NP-Hard on Relative Neighborhood Graphs and Their Relatives

**Pascal Kunz**
Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

**Till Fluschnik**
Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

**Rolf Niedermeier**
Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

**Malte Renken**
Algorithmics and Computational Complexity, Technische Universität Berlin, Germany

## Abstract

Proximity graphs have been studied for several decades, motivated by applications in computational geometry, geography, data mining, and many other fields. However, the computational complexity of classic graph problems on proximity graphs mostly remained open. We study 3-Colorability, Dominating Set, Feedback Vertex Set, Hamiltonian Cycle, and Independent Set on the following classes of proximity graphs: relative neighborhood graphs, Gabriel graphs, and relatively closest graphs. We prove that all of the aforementioned problems remain NP-hard on these graphs, except for 3-Colorability and Hamiltonian Cycle on relatively closest graphs, where the former is trivial and the latter is left open. Moreover, for every NP-hard case we additionally show that no $2^{o(n^{1/4})}$-time algorithm exists unless the Exponential-Time Hypothesis (ETH) fails, where $n$ denotes the number of vertices.

**2012 ACM Subject Classification** Theory of computation → Computational geometry; Theory of computation → Problems, reductions and completeness

**Keywords and phrases** Proximity Graphs, Relatively Closest Graphs, Gabriel Graphs, 3-Colorability, Dominating Set, Feedback Vertex Set, Hamiltonian Cycle, Independent Set, Exponential-Time Hypothesis

**Digital Object Identifier** 10.4230/LIPIcs.SWAT.2022.29

**Related Version** *Full Version*: `https://arxiv.org/abs/2107.04321` [26]

**Funding** *Pascal Kunz*: Supported by DFG Research Training Group 2434 "Facets of Complexity".
*Till Fluschnik*: Supported by DFG, projects TORE, NI 369/18, and MATE, NI 369/17.
*Malte Renken*: Supported by DFG, project MATE, NI 369/17.

**Acknowledgements** This work is based on the first author's master's thesis.
In memory of Rolf Niedermeier, our colleague, friend, and mentor, who sadly passed away before this paper was published.

## 1 Introduction

Proximity graphs describe the distance relationships between points in the plane or higher-dimensional structures. They are mostly studied in computational geometry, yet arise in numerous fields of science and engineering from geography to pattern recognition [24, 35]. In this paper, we study the computational complexity of classic NP-complete problems on three specific proximity graphs: relative neighborhood graphs (RNGs) [34], Gabriel graphs (GGs) [18], and relatively closest graphs (RCGs) [27]. All three are subgraphs of the

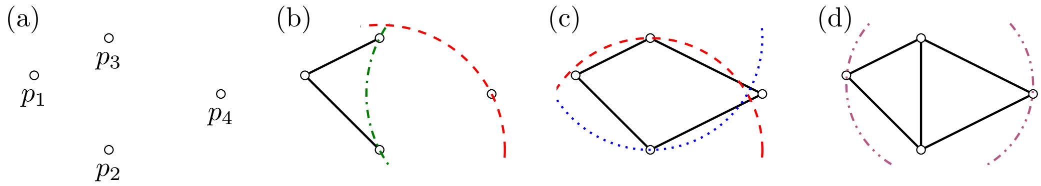


**Figure 1** (a) Set $P$ of four points $p_1 = (0, 2)$, $p_2 = (2, 0)$, $p_3 = (2, 3)$, and $p_4 = (5, 3/2)$. (b) RCG on $P$. Point $p_4$ is not adjacent with $p_2$, since $p_3$ lies in the region of influence indicated with green (dashdotted) and red (dashed). (c) RNG on $P$. Points $p_2$ and $p_3$ are not adjacent since $p_1$ is in their region of influence indicated with red (dashed) and blue (dotted). (d) GG on $P$. Points $p_1$ and $p_4$ are not adjacent since $p_2$ is in their region of influence indicated with magenta (dashdotted).

better-known Delaunay triangulation (DT). For DTs, the complexity of the restrictions of some classic NP-complete graph problems have already been resolved [12, 15] and we extend this research to these three classes.

RCGs, RNGs, and GGs are examples of empty region graphs [10]. Every pair of points is associated with a region in the plane, their region of influence, and is connected by an edge if there is no other point in that region (see Figure 1). In RNGs (RCGs), two points' region of influence is the intersection of open (closed) disks centered on each of the points with a radius equal to their distance. In a GG, two points' region of influence is a closed disk whose center is midway between them and whose diameter is their distance.

**Motivation.** Consider a railway network. It may make sense to build a track directly from one city to another, if there is no third city in between. If we interpret the area between two cities as a region of influence, then this makes proximity graphs such as RNGs plausible models for such networks. One might want to build as few maintenance facilities for the network as possible such that every track has a facility at one of its endpoints. This is an instance of the Vertex Cover (VC) problem (closely related to Independent Set (IS)). While VC is NP-hard on general graphs, one wonders whether it might be easier on proximity graphs. We will show that the problem remains NP-hard on RCGs, RNGs, and GGs. For other problems, there are similar application scenarios.

**Related Work.** Existing combinatorial results on the three graph classes include listing forbidden subgraphs, examples of graphs contained in each class, and bounds on the edge density [6, 13, 24, 30, 37]. Much algorithmic research on proximity graphs has focused on devising algorithms that efficiently compute the proximity graph from a point set (see [23] and [31] for an overview). On Delaunay triangulations, Hamiltonian Cycle is NP-hard [15], whereas 3-Colorability is polynomial-time solvable [12]. Cimikowski conjectured 3-Colorability to be NP-hard on RNGs and GGs [11]. Furthermore, he proposed a heuristic for coloring GGs and a linear-time algorithm for computing a 4-coloring in RNGs [12], but the latter has some issues, which we will discuss in Section 3.

**Our Contributions.** Table 1 summarizes our results. We prove that 3-Colorability (3-Col), Dominating Set (DS), Feedback Vertex Set (FVS), Hamiltonian Cycle (HC), and Independent Set (IS) remain NP-hard on RNGs and GGs, in particular confirming the aforementioned conjecture by Cimikowski [11]. On RCGs, 3-Colorability is trivial, but we prove that Dominating Set, Feedback Vertex Set, and Independent Set remain NP-hard. All our NP-hardness results hold true even for graphs of fairly small maximum

**Table 1** Overview of our results. Herein, $\Delta$ denotes the maximum vertex degree.
$^{\dagger}$ No $2^{o(n^{1/4})}$-time algorithm exists unless the ETH fails, where $n$ denotes the number of vertices.

| | RCGs | RNGs | GGs |
|---|---|---|---|
| 3-Colorability (3-Col) | trivial | NP-hard$^{\dagger}$, even if $\Delta = 7$ (Thm. 3.1) | |
| Dominating Set (DS) | NP-hard$^{\dagger}$, even if $\Delta = 4$ (Thm. 6.1) | | |
| Feedback Vertex Set (FVS) | NP-hard$^{\dagger}$, even if $\Delta = 4$ (Thm. 4.1) | | |
| Hamiltonian Cycle (HC) | *open* | NP-hard$^{\dagger}$, even if $\Delta = 4$ (Thm. 5.1) | |
| Independent Set (IS) | NP-hard$^{\dagger}$, even if $\Delta = 4$ (Thm. 6.1) | | |

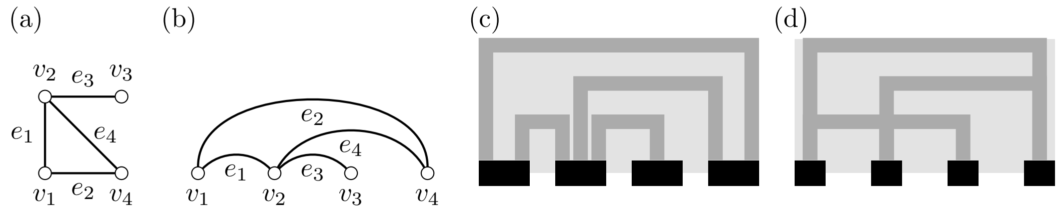


**Figure 2** (a) A graph $G$ with vertex set $\{v_1, \ldots, v_4\}$. (b) A 1-page book embedding of $G$. (c) and (d) Illustration of our technique, where black rectangles correspond to vertex gadgets, thick gray lines to edge/connector gadgets, and light gray areas indicate filler gadgets.

degree (7 in the case of 3-Col, and 4 in all other cases, yielding a dichotomy between polynomial-time solvability and NP-hardness in the case of FVS). We complement each NP-hardness result with a running-time lower bound of $2^{o(n^{1/4})}$ based on the Exponential-Time Hypothesis, where $n$ is the number of vertices. The fastest known algorithms for these problems run in time $2^{O(n^{1/2})}$ on planar graphs and it remains open whether a running time between these lower and upper bounds can be achieved on proximity graphs. Many details and proofs (marked with ★) are deferred to the full version of this paper.

**Our Technique.** In our NP-hardness proofs (see Table 1), we give polynomial-time many-one reductions from each problem's restriction to planar graphs with maximum degree 3 or 4. We proceed as follows (see Figure 2 for an illustration). We exploit the fact that for any planar graph with maximum degree at most 4, we can compute in polynomial time a 2-page book embedding [3], a very structured representation of the input graph. Then, we translate the book embedding's structure into a grid-like structure. Each reduction uses three types of gadgets: to represent vertices, to represent edges, and to fill the space between them in order to prevent the appearance of unwanted edges between the other gadgets.

## 2 Preliminaries

Let $\mathbb{N} := \{1, 2, 3, \ldots\}$ and $\mathbb{N}_0 := \{0\} \cup \mathbb{N}$. We use basic notions from graph theory [14].

**Proximity graphs.** Let $\mathrm{d}\colon \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R}$ denote the Euclidean distance between two points. The *open* and *closed ball* with radius $r > 0$ and center $p \in \mathbb{R}^2$ are

$$B_r(p) := \{q \in \mathbb{R}^2 \mid \mathrm{d}(p, q) < r\} \text{ and } \overline{B}_r(p) := \{q \in \mathbb{R}^2 \mid \mathrm{d}(p, q) \leq r\}.$$

The Delaunay triangulation of $P \subseteq \mathbb{R}^2$ (see, e.g., [4, Ch. 9]) is denoted by $\mathrm{DT}(P)$.

A *template region* [10] is a function $R\colon \binom{\mathbb{R}^2}{2} \to 2^{\mathbb{R}^2}$ that assigns a region of the plane, called the *region of influence*, to each pair of points in the plane. Given a template region $R_{\mathcal{C}}$ and points $p_1, p_2 \in \mathbb{R}^2$, a third point $p_3 \in \mathbb{R}^2 \setminus \{p_1, p_2\}$ is a *$\mathcal{C}$-blocker* for $\{p_1, p_2\}$ if $p_3 \in R_{\mathcal{C}}(p_1, p_2)$. For a finite set of points $P = \{p_1, \dots, p_n\} \subseteq \mathbb{R}^2$, the $\mathcal{C}$-graph *induced* by $P$ is $\mathcal{C}(P) \coloneqq (\{v_1, \dots, v_n\}, E_{\mathcal{C}(P)})$ with

$$E_{\mathcal{C}(P)} \coloneqq \{\{v_i, v_j\} \mid (R_{\mathcal{C}}(p_i, p_j) \cap P) \setminus \{p_i, p_j\} = \emptyset \text{ and } i \neq j\}.$$

The class $\mathcal{C}$ contains a graph $G$ if there is a finite set of points $P \subseteq \mathbb{R}^2$ with $\mathcal{C}(P) = G$.

We are interested in three template regions and the graph classes defined by them:

***Relatively closest graphs (RCGs)*:** Defined by

$$\begin{aligned} R_{\mathrm{RCG}}(p_1, p_2) &\coloneqq \{p_3 \in \mathbb{R}^2 \mid \mathrm{d}(p_1, p_2) \geq \max\{\mathrm{d}(p_1, p_3), \mathrm{d}(p_2, p_3)\}\} \\ &= \overline{B}_{\mathrm{d}(p_1,p_2)}(p_1) \cap \overline{B}_{\mathrm{d}(p_1,p_2)}(p_2). \end{aligned}$$

***Relative neighborhood graphs (RNGs)*:** Defined by

$$\begin{aligned} R_{\mathrm{RNG}}(p_1, p_2) &\coloneqq \{p_3 \in \mathbb{R}^2 \mid \mathrm{d}(p_1, p_2) > \max\{\mathrm{d}(p_1, p_3), \mathrm{d}(p_2, p_3)\}\} \\ &= B_{\mathrm{d}(p_1,p_2)}(p_1) \cap B_{\mathrm{d}(p_1,p_2)}(p_2). \end{aligned}$$

***Gabriel graphs (GGs)*:** Defined by

$$R_{\mathrm{GG}}(p_1, p_2) \coloneqq \{p_3 \in \mathbb{R}^2 \mid \mathrm{d}(p_1, p_2)^2 \geq \mathrm{d}(p_1, p_3)^2 + \mathrm{d}(p_2, p_3)^2\} = \overline{B}_{\mathrm{d}(p_1,p_2)/2}(q),$$

where $q$ is the midpoint between $p_1$ and $p_2$.

A *$\mathcal{C}$-embedding* of a graph $G = (V, E)$ is a mapping $\mathrm{emb}\colon V \to \mathbb{R}^2$ such that $\mathcal{C}(\mathrm{emb}(V)) = G$.[1] Of course, $G \in \mathcal{C}$ if and only if $G$ admits a $\mathcal{C}$-embedding.

For any finite point set $P$, it holds that $E_{\mathrm{RCG}(P)} \subseteq E_{\mathrm{RNG}(P)} \subseteq E_{\mathrm{GG}(P)} \subseteq \mathrm{DT}(P)$ [13]. RCGs cannot contain $K_3$ as a subgraph and none of the three can contain $K_4$ or $K_{2,3}$ [13, 30, 37]. Moreover, $p_3 \in R_{\mathrm{GG}}(p_1, p_2)$ if and only if the angle at $p_3$ formed by the lines to $p_1$ and $p_2$ is at most 90° [30]. Finally, the following lemma will be used to prove that graphs are in each graph class:

▶ **Lemma 2.1** ([30])**.** *Let $P$ be a set of points in the plane and $G$ the RCG, RNG, or GG induced by $P$. Then, the straight-line drawing of $G$ induced by $P$ is planar.*

**Book embeddings.** Our NP-hardness proofs use 2-page book embeddings. A *$k$-page book embedding* of a graph $G = (V, E)$ consists of

**(i)** an edge partition $E = E_1 \uplus \cdots \uplus E_k$, and

**(ii)** for every $i \in \{1, \dots, k\}$, a planar embedding $\mathrm{emb}_i$ of $(V, E_i)$ in $\mathbb{R} \times \mathbb{R}_{\geq 0}$, where $\mathrm{emb}_i(v) = \mathrm{emb}_j(v) \in \mathbb{R} \times \{0\}$ for every $v \in V$, $i, j \in \{1, \dots, k\}$.

The following result due to Bekos et al. [3] will play an important role in this work:

▶ **Theorem 2.2** ([3])**.** *Every planar graph with maximum degree at most 4 admits a 2-page book embedding. Such an embedding can be computed in quadratic time.*

The following terminology will be useful in our NP-hardness proofs. Consider a graph $G = (V, E)$ and a 2-page book embedding of $G$. Let $v_1, \dots, v_n$ be the vertices of the graph ordered in such a way that $\mathrm{emb}_i(v_j) < \mathrm{emb}_i(v_{j+1})$ for $i \in \{1, 2\}$ and every $j \in \{1, \dots, n-1\}$. We

[1] To simplify notation, we write $v$ instead of $\mathrm{emb}(v)$ to refer to the point at which vertex $v$ is embedded.

will say that $v_1, \dots, v_n$ is the *order in which the vertices appear on the spine*. Let $r \in \{1,2\}$. We will use $N_r(v) \coloneqq \{v' \mid \{v,v'\} \in E_r\}$ to denote $v$'s *$E_r$-neighborhood* and $\deg_r(v) \coloneqq |N_r(v)|$ to denote $v$'s *$E_r$-degree*. For an edge $e = \{v_i, v_j\}$, $i < j$, define its *length* as $\ell(e) \coloneqq j - i$. The *interior* of $e \in E_r$ is

$$\operatorname{int}(e) \coloneqq \{e' = \{v_{i'}, v_{j'}\} \in E_r \mid i \le i' < j' \le j, e' \neq e\}.$$

The *height* of $e$ is $h(e) \coloneqq 1 + \max\{0, h(e') \mid e' \in \operatorname{int}(e)\}$. Note that, because the height of any edge only depends on the height of shorter edges, edge height is well-defined. The length and height of an edge are both in $\mathcal{O}(n)$. The *$E_r$-height* of a vertex $v$ is

$$h_r(v) \coloneqq \max\{0, h(e) \mid v \in e \in E_r\}, \quad \text{where} \quad h(e) \coloneqq 1 + \max\{0, h(e') \mid e' \in \operatorname{int}(e)\}.$$

Let $h_r(G) \coloneqq \max\{h_r(v_i) \mid i \in \{1,\dots,n\}\}$. For every vertex $v_i$, we order its incident edges in $E_r$ as follows. If $N_r(v_i) = \{v_{j_1}, \dots, v_{j_k}\}$ with $j_1 < \dots < j_c < i < j_{c+1} < \dots < j_k$, then the order of the edges is $\{v_i, v_{j_c}\} < \dots < \{v_i, v_{j_1}\} < \{v_i, v_{j_k}\} < \dots < \{v_i, v_{j_{c+1}}\}$.

**Grid structure.** The graphs we build in our reductions will all have a grid-like structure. With the exception of the reduction for Hamiltonian Cycle, we group their vertices into *$(x,y)$-corners* with $(x,y) \in \mathbb{Z}^2$ and there will be a corner for every $(x,y)$ within certain bounds, which depend on the problem in question as well as the size and structure of the input graph. In the embedding, the vertices forming the $(x,y)$-corner will be in $\overline{B}_r(x,y)$ for a suitable $r > 0$. Some vertices are not part of any corner and are called *intermediate vertices*. They are usually located midway between two corners. Each corner can have one or multiple dedicated right, top, left, and bottom *connecting vertices*. If a corner consists of a single vertex, that vertex always simultaneously acts as the right, top, left, and bottom connecting vertex of that corner. The connecting vertices of a corner are the only ones that may have neighbors outside of that corner. For any $(x,y) \in \mathbb{Z}^2$, we say that the vertices in the $(x,y)$, $(x+1,y)$, $(x,y+1)$, and $(x+1,y+1)$-corners along with any intermediate vertices that are adjacent to vertices in two of the aforementioned corners jointly form a *grid face*.

**Exponential-Time Hypothesis.** The *Exponential-Time Hypothesis (ETH)*, introduced by Impagliazzo and Paturi [22], has become a useful tool to give more precise running-time lower bounds than the more classical dichotomy between polynomial-time solvable and NP-hard problems (for an overview, we refer to [28]). This conjecture states:

▶ **Hypothesis 2.3** (Exponential-Time Hypothesis [22])**.** *There is some fixed $c > 0$ such that* 3-CNF-Sat *is not solvable in $2^{cn} \cdot (n+m)^{\mathcal{O}(1)}$ time, where $n$ and $m$ denote the numbers of variables and clauses, respectively.*

## 3 3-Colorability

We start with the 3-Colorability problem. A *3-coloring* of a graph $G = (V,E)$ is a function $c\colon V \to \{1,2,3\}$ such that $c(u) \neq c(v)$ for all $\{u,v\} \in E$. In the 3-Colorability problem (3-Col), one is given a graph as input and asked to decide whether it admits a 3-coloring. Every RCG is 3-colorable [13] because RCGs do not contain any 3-cycles and every planar graph without 3-cycles is 3-colorable by Grötzsch's theorem [21]. As a result, 3-Col is trivial when restricted to RCGs. Regarding RNGs and GGs, we prove the following.

▶ **Theorem 3.1** (★)**.** 3-Colorability *on RNGs and on GGs is* NP*-hard, even if the maximum degree is 7. Moreover, unless the ETH fails, it admits no $2^{o(n^{1/4})}$-time algorithm where $n$ is the number of vertices.*

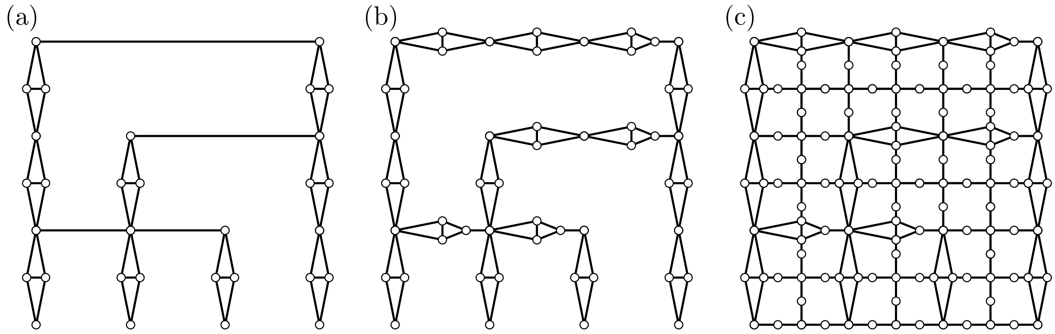


**Figure 3** The construction proving Theorem 3.1 applied to the example graph from Figure 2. (a) The graph after steps 1–2. (b) The graph after steps 1–3. (c) The final graph.

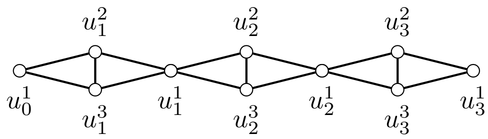


**Figure 4** A coloring path of length 3.

This confirms a conjecture by Cimikowski [11]. Our proof is based on a polynomial-time many-one reduction from the NP-hard [19] 3-Colorability problem on planar graphs with maximum degree 4. The complete proof is deferred to the full version, but we give the construction and some intuition concerning its correctness. We give an illustration of the construction in Figure 3 and the following high-level description: After computing a 2-page book embedding, we replace each vertex by a "color-preserving" path (Figure 3(a)). We further replace each original edge by a path enforcing different colors for the respective endpoints (Figure 3(b)). Finally, we fill the remaining space with a subdivided grid (Figure 3(c)), which preserves 3-colorability and turns the graph simultaneously into an RNG and a GG.

We will use so-called *coloring paths* (see Figure 4 for an illustration), which essentially allows us to copy the color of a vertex. The *coloring path* of length $k$ from $u_0^1$ to $u_k^1$ is the graph $\tilde{P}_k \coloneqq (V_k, E_k)$ with:

$$V_k \coloneqq \{u_0^1\} \cup \{u_i^1, u_i^2, u_i^3 \mid i \in \{1, \dots, k\}\} \text{ and}$$
$$E_k \coloneqq \{\{u_{i-1}^1, u_i^2\}, \{u_{i-1}^1, u_i^3\}, \{u_i^2, u_i^3\}, \{u_i^2, u_i^1\}, \{u_i^3, u_i^1\} \mid i \in \{1, \dots, k\}\}.$$

We will call $u_i^1$ the $i$-th *center vertex*, $u_i^2$ the $i$-th *left vertex*, and $u_i^3$ the $i$-th *right vertex*.

▶ **Construction 3.2.** Let $G = (V, E)$ be an undirected planar graph of maximum degree 4. We will construct a graph $G' = (V', E')$ and subsequently show that $G'$ is both an RNG and a GG and that $G$ is 3-colorable if and only if $G'$ is. The vertex set of $G'$ will mostly consist of groups of vertices called $(x, y)$-*corners* where $2 \leq x \leq 2n$ and $-2h_2(G) \leq y \leq 2h_1(G)$. Each corner consists of either a single vertex or of a pair of adjacent vertices. Corners can have dedicated top, left, right, and bottom *connecting vertices*, some of which may coincide. For example, if a corner consists of a single vertex, that vertex simultaneously forms all four connecting vertices. Finally, there will be some *intermediate vertices* that are not part of any corner.

We start with $G' \coloneqq G$.

**Step 1:** Compute a 2-page book embedding of $G$. Let $v_1, \ldots, v_n$ be the vertices of $G$ enumerated in the order in which they appear on the spine, and let $\{E_1, E_2\}$ denote the partition of $E$.

**Step 2:** Replace every vertex $v_i$ with a coloring path of length $h_1(v_i) + h_2(v_i)$. Every edge $e$ of $G$ incident to $v_i$ is now instead attached to the $(h_2(v_i)+h(e))$-th center vertex, if $e \in E_1$, or the $(h_2(v_i) - h(e))$ if $e \in E_2$. For $r = 0, \ldots, h_1(v_i) + h_2(v_i)$, the $r$-th center vertex of that path forms the $(2i, 2r - 2h_2(v_i))$-corner. For $r = 1, \ldots, h_1(v_i)$, the $r$-th left and right vertices jointly form the $(2i, 2r - 1 - 2h_2(v_i))$-corner. The left vertex is the left connecting vertex of this corner and the right vertex is the right connecting vertex.

**Step 3:** For every edge $e = \{v_i, v_j\} \in E_1$, $i < j$, replace the corresponding edge of $G'$ with a coloring path of length $\ell(e)$. Identify the first vertex of that path with the $(2i, 2h(e))$-corner (which consists of a single vertex). Denote the last vertex of that path by $w$. Add an edge from $w$ to the $(2j, 2h(e))$-corner. For $r = 1, \ldots, \ell(e) - 1$, the $r$-th center vertex of that path is the $(2i + 2r, 2h(e))$-corner. The vertex $w$, which is the $\ell(e)$-th center vertex, is an intermediate vertex. For $r = 1, \ldots, \ell(e)$, the $r$-th left and right vertices jointly form the $(2i + 2r - 1, 2h(e))$-corner. The left vertex is the top connecting vertex of this corner and the right vertex is the bottom connecting vertex.

**Step 4:** For every $(x, y)$ with $2 \le x \le 2n$ and $0 \le y \le 2h_1(G)$, if an $(x, y)$-corner was not added in one of the previous two steps, then add a single vertex, which becomes the $(x, y)$-corner, to $G'$. In that case add an edge from the $(x, y)$-corner to the top connecting vertex of the $(x, y - 1)$-corner, to the left connecting vertex of the $(x + 1, y)$-corner, and so on. Subdivide each of these edges once, introducing four new intermediate vertices.

Note that Steps 3 and 4 take only $E_1$ into account. These steps must be repeated analogously for $E_2$, using negative $y$-coordinates. ⌟

The correctness builds on the following two facts. By replacing each original edge with a path of coloring paths in Steps 2 and 3, we enforce different colors for the respective endpoints (see Figure 3(b)). Filling the remaining space with a subdivided grid in Step 4 (see Figure 3(c)) preserves 3-colorability and turns the graph both into an RNG and a GG.

**Further remarks.** We remark that RNGs and GGs with maximum degree 3 are always 3-colorable. This follows from Brooks' theorem [9, 29], which states that any graph with maximum degree $\Delta \ge 3$ is $\Delta$-colorable, if it contains no $(\Delta + 1)$-clique. RNGs and GGs contain no 4-cliques (see Section 2). It remains open whether 3-Col can be solved in polynomial time on RNGs or GGs with maximum degree between 4 and 6.

By the well-known four color theorem, all planar graphs are 4-colorable. The fastest known algorithm to compute a 4-coloring of a planar graph has quadratic running time [32]. For RNGs, Cimikowski [12] proposed an algorithm for computing 4-colorings in linear time. However, this algorithm is based on the claim [37, Lemma 4.2] that the wheel graph $W_6$ cannot occur as subgraph of an RNG. This claim was disproved by Bose et al. [6]. Cimikowski's algorithm additionally implicitly assumes that RNGs are closed under minors, since the algorithm sometimes merges two adjacent vertices. This can lead to graphs that are not RNGs. Thus, it remains open whether or not a linear-time algorithm for this task exists.

## 4 Feedback Vertex Set

We now turn our attention to the Feedback Vertex Set problem. In a graph $G = (V, E)$, a *feedback vertex set* is a set $X \subseteq V$ such that $G - X$ is a forest. In the Feedback Vertex Set problem (FVS), one is given a graph and $k \in \mathbb{N}_0$ and asked whether the graph contains a feedback vertex set of size at most $k$. We prove the following:

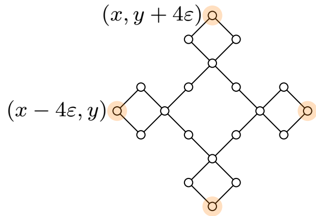


**Figure 5** Buffer at position $(x, y) \in \mathbb{R}^2$ with highlighted outer vertices.

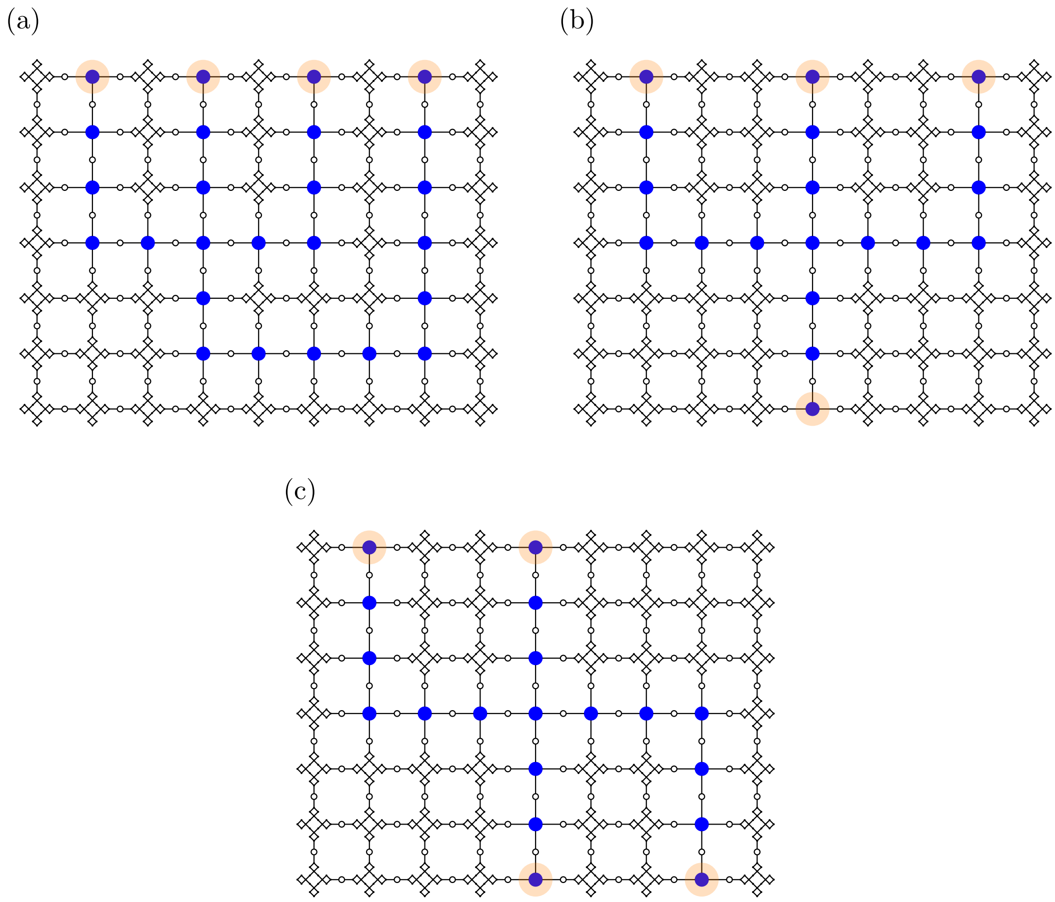


**Figure 6** The (a) $(4, 0)$-, (b) $(3, 1)$-, and (c) $(2, 2)$-vertex gadget with highlighted outlets.

▶ **Theorem 4.1 (★).** Feedback Vertex Set *on RCGs, on RNGs, and on GGs is* NP*-hard, even if the maximum degree is* 4*. Moreover, unless the ETH fails, it admits no* $2^{o(n^{1/4})}$*-time algorithm where n is the number of vertices.*

Our proof is based on a polynomial-time many-one reduction from the NP-complete [33] Feedback Vertex Set on planar graphs of maximum degree 4. We only give an intuitive high-level description here, deferring the complete proof to the full version, and illustrate the resulting graph in Figure 7. We utilize the graph pictured in Figure 5 and call it a "buffer". Adding a copy of this graph to an existing graph increases the size of a minimum feedback vertex set by exactly 4 even if the vertices marked as outer vertices are each connected to exactly one vertex in the original graph.

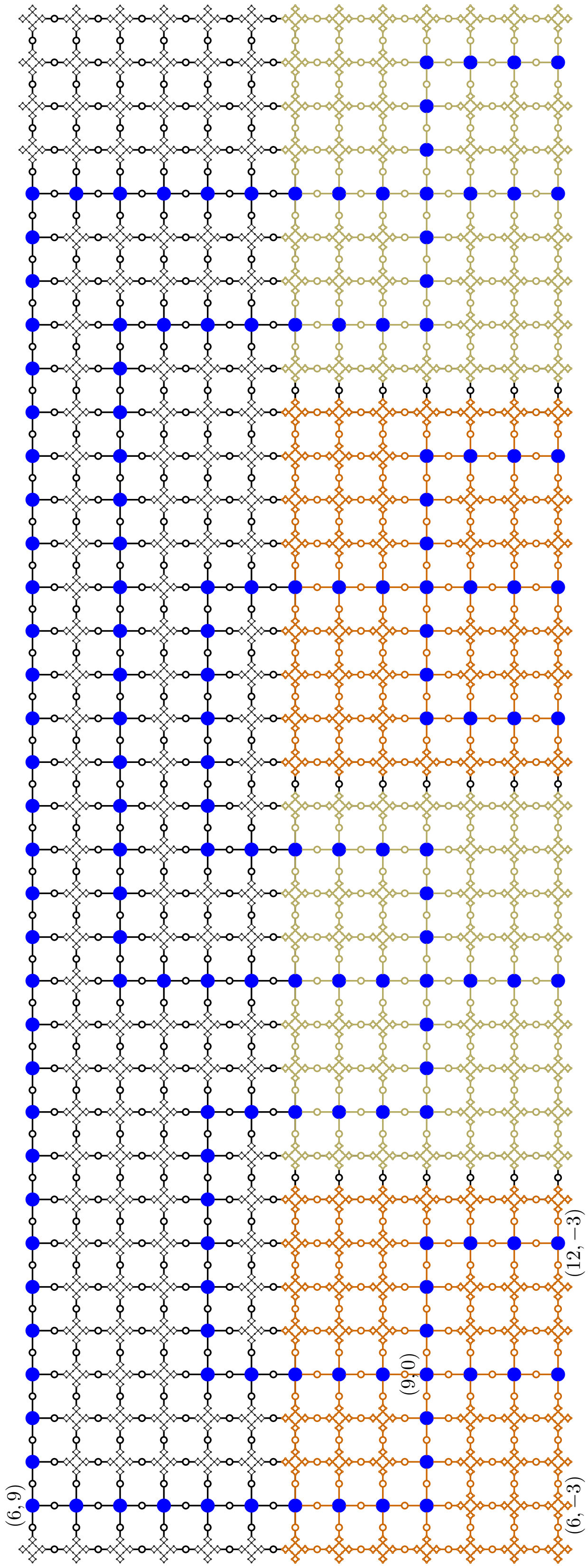


**Figure 7** The construction proving Theorem 4.1 applied to the graph in Figure 2. The large blue vertices along with the degree-2 vertices between them form a subdivision of the original graph (with a forest added). The bottom half consists of four vertex gadgets, colored alternatingly orange and yellow.

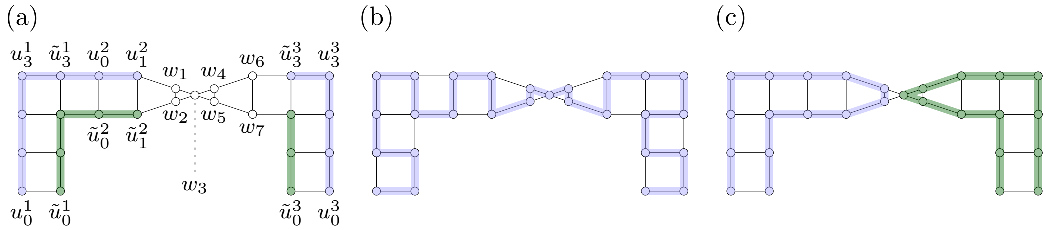


**Figure 8** The ladder path $L_{4,2}$ with (a) selected vertex labels and inside/outside edges highlighted in dark green / light blue; (b) a traversal; (c) a partial/full cover (light blue / dark green).

We compute a 2-page book embedding of the input graph and represent every vertex $v$ in that graph by a gadget which depends on $\deg_1(v)$ and $\deg_2(v)$. The three possible gadgets for a vertex with degree 4 are pictured in Figure 6. They contain many copies of the buffer graph. We then represent edges in the input path by long paths between the outlets of the vertex gadgets that represent the edges' endpoints. We fill the space between these paths with copies of the buffer.

The resulting graph has the following structure once all buffers have been deleted (this is the blue subgraph in Figure 7). Iteratively removing degree-1 vertices from this graph results in a subdivision of the input graph. This leads to our correctness proof for this reduction, as FVS is invariant under deletion of degree-1 vertices and under subdivisions.

Note that FVS is polynomial-time solvable on graphs with maximum degree 3 [36].

## 5 Hamiltonian Cycle

In a graph $G = (V, E)$, a *Hamiltonian cycle* is a cycle that visits every vertex in $V$ exactly once. In the Hamiltonian Cycle problem (HC), one is given a graph and asked to decide whether the graph contains a Hamiltonian cycle. As the proof of the following is more complex than for previous problems, we will give more details.

▶ **Theorem 5.1.** Hamiltonian Cycle *on RNGs and on GGs is* NP*-hard, even if the maximum degree is* 4*. Moreover, unless the ETH fails, it admits no* $2^{o(n^{1/4})}$*-time algorithm where* $n$ *is the number of vertices.*

To prove Theorem 5.1, we give a polynomial-time many-one reduction from the restriction of Hamiltonian Cycle to 3-regular planar graphs, for which we have the following.

▶ **Proposition 5.2** ([20, 28])**.** Hamiltonian Cycle *on* 3*-regular planar graphs is* NP*-hard and, unless the ETH fails, admits no* $2^{o(n^{1/2})}$*-time algorithm where* $n$ *is the number of vertices.*

The reduction in the proof of Theorem 5.1 consists of two Hamiltonicity-preserving modifications: gadget expansion (Section 5.1) and face filling (Section 5.2).

### 5.1 Gadget Expansion

The gadgets that will replace the edges are called *ladder paths.* For $k_1, k_2 \in \mathbb{N}$, the *ladder path* with length $(k_1, k_2)$ is the graph $L_{k_1,k_2} = (V, E)$, where

$$V = \{u_i^1, \tilde{u}_i^1, u_i^3, \tilde{u}_i^3 \mid i \in \{0, \dots, k_1 - 1\}\} \cup \{u_i^2, \tilde{u}_i^2 \mid i \in \{0, \dots, k_2 - 1\}\} \cup \{w_1, \dots, w_7\}.$$

The edges are given using the example pictured in Figure 8(a) and listed explicitly in the full version.

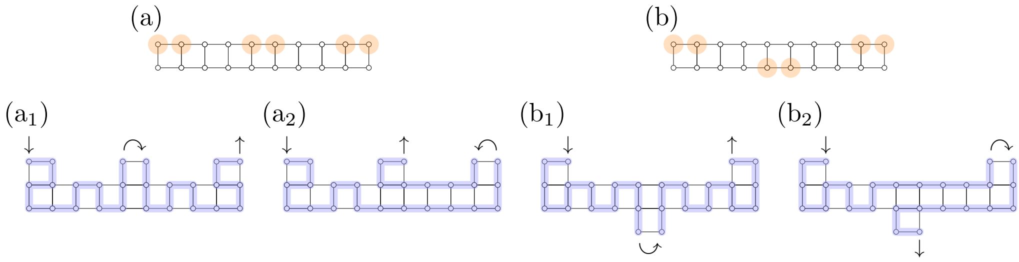


**Figure 9** (a) $(3,0)$-vertex gadget and (b) $(2,1)$-vertex gadget, with highlighted outlets. (a$_1$), (a$_2$) and (b$_1$), (b$_2$) show two generic ways a Hamiltonian cycle can pass through each vertex gadget.

The vertices $u_i^j, \tilde{u}_i^j$ with $j \in \{1,2\}$ along with $w_1$ and $w_2$ form the *first half* of the ladder path and those with $j = 3$ along with $w_4, \ldots, w_7$ form the *second half*. The vertex $w_3$ is the transitional vertex. The vertices $w_1, \ldots, w_7$ form the *switch*. The vertices $u_0^1$ and $\tilde{u}_0^1$ form the *end* of the first half, while $u_0^3$ and $\tilde{u}_0^3$ form the *end* of the second half. The edges highlighted in light blue in Figure 8(a) will be called *outside edges*, while the edges highlighted in dark green are *inside edges*. An edge $\{u_i^j, u_{i+1}^j\}$ or $\{\tilde{u}_i^j, \tilde{u}_{i+1}^j\}$ is called *even* if $i$ is even.

A *traversal* of a ladder path is a path that begins in either vertex at one end of the ladder path, terminates in either vertex at the other end, and visits every vertex on the ladder path and no other vertex. A *partial cover* of a half of a ladder path is a path that begins in either vertex in the end of the half, terminates in the other vertex in that end, and visits every vertex of that half, but no other vertex. A *full cover* of a half additionally visits the transitional vertex. Examples of a traversal, a partial cover, and a full cover are pictured in Figure 8(b) and (c). The main property of ladder paths is that any Hamiltonian cycle must either contain a traversal, or a full and a half cover of each ladder path.

▶ **Lemma 5.3** (★). *Suppose that the Hamiltonian graph $G = (V, E)$ contains a ladder, that the only vertices on the ladder path with neighbors outside of the ladder path are on its ends, and that the vertices on the ends each have no more than one neighbor outside of the ladder path. Then, any Hamiltonian cycle in $G$ contains either:*

- *a traversal of the ladder path or*
- *a partial cover of one of its halves and a full cover of the other half.*

For a ladder path and a Hamiltonian cycle in a graph, we say that the ladder path is *traversed* if the Hamiltonian cycle contains a traversal of the ladder path. Otherwise, it is *covered*.

Next, we discuss the vertex gadgets. Recall that the graph $G$ is assumed to be 3-regular. We will use four types of vertex gadgets. Each vertex gadget consist of a grid of size $2 \times 10$, with the only difference being the position of their three *outlets*, which are designated vertex pairs to which the ladder paths representing the edges will be connected. The $(3,0)$-*vertex gadget* and the $(2,1)$-*vertex gadget* are pictured in Figure 9 with the three outlets highlighted. The $(0,3)$-*vertex gadget* and the $(1,2)$-*vertex gadget* are obtained from the former two by mirroring along the horizontal axis. The value $(i, j)$ will be called the *type* of the gadget.

We will refer to the outlets as *top* or *bottom* outlets, as well as the *left*, *middle* or *right* outlet, with the obvious meaning. The left and right outlets are also called the *outer outlets*.

We will now define the *gadget expansion* of a 3-regular graph $G$, consisting of a graph $G'$ and a straight-line embedding emb resulting from applying the following steps to $G$.

▶ **Construction 5.4** (Gadget expansion). Start with $G'$ being the empty graph.

**Step 1:** Compute a 2-page book embedding of $G$ and let $v_1, \ldots, v_n$ be the vertices of $G$ in the order in which they appear on the spine, and let $\{E_1, E_2\}$ denote the partition of $E$.

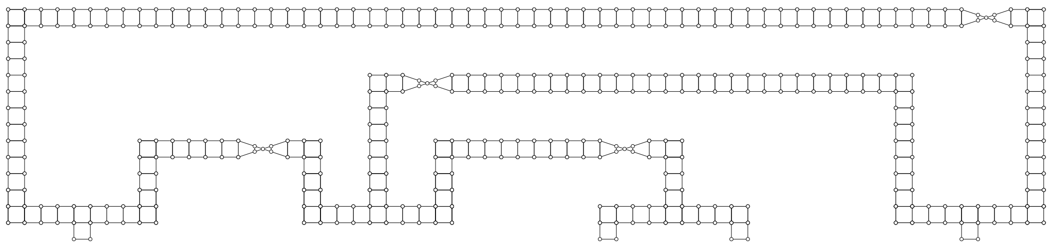

**Figure 10** Gadget expansion of the graph pictured in Figure 2. This graph is not 3-regular, but we may assume that there are further edges in $E_2$.

**Step 2:** For every vertex $v_i \in V$ add to $G'$ a $(\deg_1(v_i), \deg_2(v_i))$-vertex gadget. Position the vertices of this gadget at $(18i + x, y)$ with $x \in \{0, \ldots, 9\}$ and $y \in \{0, 1\}$, as in Figure 9.

**Step 3:** For every edge $e = \{v_i, v_j\}$ in $E_1$, $i < j$, add to $G'$ a ladder path $L_{k_1,k_2}$ connected to an outlet in $v_i$'s vertex gadget and an outlet in $v_j$'s vertex gadget as follows. Recall the ordering of the edges incident to a vertex defined in the preliminaries. If $e$ is the $r$-th edge at $v_i$ and the $s$-th edge at $v_j$, then attach said ladder path to the $r$-th top outlet from the left of $v_i$'s vertex gadget and to the $s$-th top outlet from the left of $v_j$'s vertex gadget. If only one of these two outlets is an outer outlet, then attach the end of the first half to that outlet and the end of the second half to the other (middle) outlet. (If the outlets are both outer or both middle outlets, then it does not matter which end of the ladder path is connected to which outlet.) This is done by adding two disjoint edges which connect the two vertices forming an end of the ladder path to the two vertices forming the corresponding outlet as in Figure 10.

The value of $k_1$ is chosen as $k_1 \coloneqq 6h(e)$ and the value of $k_2$ as follows. Define $\alpha$ to be 0, 4, or 8, if the ladder path is attached to the left, middle, or right outlet of $v_i$'s vertex gadget, respectively. Define $\beta$ in the same manner for $v_j$. Set $k_2 \coloneqq 18(j - i) - \alpha + \beta - 5$. Note that $k_2 \geq 5$ always holds. Finally, we will give the embedding of the ladder path's vertices, using the designations introduced in the definition of a ladder path. We only state the case where the first half is attached to $v_i$, for the other case the coordinates are to be mirrored at a suitable vertical axis. The positions of the vertices in the first half are (with $S_i^\alpha \coloneqq 18i + \alpha$):

$$
\begin{aligned}
&\mathrm{emb}(u_r^1) \coloneqq (S_i^\alpha, r + 2), && \mathrm{emb}(\tilde{u}_r^1) \coloneqq (S_i^\alpha + 1, r + 2), \quad r = 0, \ldots, k_1 - 1,\\
&\mathrm{emb}(u_r^2) \coloneqq (S_i^\alpha + 2 + r, k_1 + 1), && \mathrm{emb}(\tilde{u}_r^2) \coloneqq (S_i^\alpha + 2 + r, k_1), \quad r = 0, \ldots, k_2 - 1,\\
&\mathrm{emb}(w_1) \coloneqq (S_i^\alpha + k_2 + 2, k_1 + 2/3), && \mathrm{emb}(w_2) \coloneqq (S_i^\alpha + k_2 + 2, k_1 + 1/3),\\
&\mathrm{emb}(w_3) \coloneqq (S_i^\alpha + k_2 + 2.5, k_1 + 1/2), && \mathrm{emb}(w_4) \coloneqq (S_i^\alpha + k_2 + 3, k_1 + 2/3),\\
&\mathrm{emb}(w_5) \coloneqq (S_i^\alpha + k_2 + 3, k_1 + 1/3), && \mathrm{emb}(w_6) \coloneqq (S_i^\alpha + k_2 + 4, k_1 + 1),\\
&\mathrm{emb}(w_7) \coloneqq (S_i^\alpha + k_2 + 4, k_1).
\end{aligned}
$$

This embedding is illustrated in Figure 8. The positions of the vertices in the second half are analogous to the first. This step is illustrated in Figure 10.

Step 3 must be repeated for $E_2$ using negative $y$-coordinates. ⌟

This construction is useful due to the following:

▶ **Lemma 5.5 (★).** *Gadget expansion preserves Hamiltonicity.*

The proof, which is deferred to the full version, is based on the fact that a Hamiltonian cycle may pass through vertex gadgets as pictured in Figure 9.

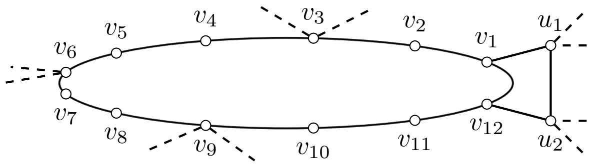


**Figure 11** An example of a permissible cycle addition with a cycle of length 12.

## 5.2 Face Filling

In order to turn the gadget expansion of a graph into an RNG and GG, we need to add *buffers*. The challenge is doing this in a way that preserves Hamiltonicity. We call an edge $e$ of $G$ *permissible* if $G$ is not Hamiltonian or if $G$ contains a Hamiltonian cycle that passes through $e$.

▶ **Lemma 5.6 (★).** *Subdividing a permissible edge preserves Hamiltonicity. Moreover, both edges resulting from the subdivision are permissible in the resulting graph.*

Our main tool for adding buffers to a graph is called *permissible cycle addition*. Let $\{u_1, u_2\}$ be a permissible edge of $G = (V, E)$. We say that $G' = (V', E')$ is obtained from $G$ by *attaching a permissible cycle to* $\{u_1, u_2\}$ if (see Figure 11 for an illustration)

- $V' = V \uplus \{v_1, \dots, v_k\}$, $k \geq 4$, and $v_1, \dots, v_k$ induce a cycle in that order, that is, $\{v_i, v_j\} \in E'$ if and only if $|i - j| = 1$ or $\{i, j\} = \{1, k\}$;
- $E' \cap \binom{V}{2} = E$;
- for all $i \in \{2, \dots, k-1\}$, if $\deg_{G'}(v_i) \geq 3$, then $\deg_{G'}(v_{i-1}) = \deg_{G'}(v_{i+1}) = 2$; and
- $\deg_{G'}(v_1) = \deg_{G'}(v_k) = 3$, and $\{\{v_1, u_1\}, \{v_k, u_2\}\} \subseteq E'$.

Figure 11 pictures an example of such a cycle addition. This modification is useful due to:

▶ **Lemma 5.7 (★).** *Permissible cycle addition preserves Hamiltonicity. Moreover, if $v_1, \dots, v_k$ is the added cycle, then the edges $\{v_i, v_{i+1}\}$, $1 \leq i < k$, are all permissible in the resulting graph.*

In order to be able to apply *permissible cycle addition* to the gadget expansion $G'$ of a graph $G$, we need to know permissible edges of $G'$. For this, we have the following lemma.

▶ **Lemma 5.8 (★).** *Let $G$ be a 3-regular graph, $G'$ the gadget expansion of $G$, and $L$ any ladder path of $G'$ whose first half is attached to an outer outlet (of a vertex gadget). Then, $L$ contains two even inside and two even outside edges, all of which are permissible. Furthermore, these edges can be determined in linear time.*

We are set to give the construction in our polynomial-time many-one reduction from HC on 3-regular planar graphs to HC on RNGs or GGs.

▶ **Construction 5.9.** Let $G = (V, E)$ be a 3-regular planar graph. We will construct an RNG and GG $G' = (V', E')$ that is Hamiltonian if and only if $G$ is. We will give the embedding of the vertices directly in the reduction.

We start with $(G', \text{emb})$, the gadget expansion of $G$. We add one buffer for every $(x, y) \in \mathbb{Z}^2$ where $14 \leq x \leq 16n + 10$ and $-6h_2(G) - 2 \leq y \leq 6h_1(G) + 2$ are both even, except when $G'$ already contains a vertex $v$ with $\text{emb}(v) \in \{(x, y), (x+1, y)\}$. The buffer then consists of a 4-cycle whose vertices are embedded at $(x, y), (x+1, y), (x+1, y+1)$, and $(x, y+1)$ and whose edges are then further subdivided. We call it the $(x, y)$-buffer and refer to the four (subdivided) edges as its *sides*. For each side, if $G'$ previously already contained

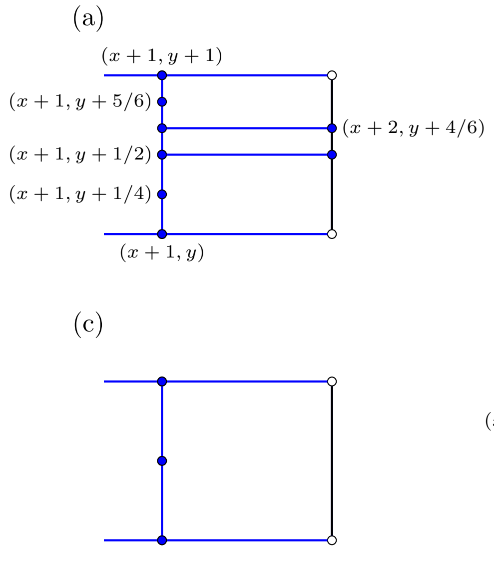


**Figure 12** Construction of the sides of a cycle (left) with (a) a docking side adjoined to an edge, (b) a docking side adjoined to a previously existing non-docking side, (c) a non-docking side adjoined to an edge, and (d) a non-docking side adjoined to an existing side. In each picture, an edge or previously existing side to adjoin is on the right. The vertices and edges that result from the addition of the cycle are marked in blue while previously existing vertices and edges are in black.

a (possibly subdivided) edge running parallel to that side at distance 1, then we say that this side *adjoins* that (possibly subdivided) edge. For example, the side from $(x, y+1)$ to $(x+1, y+1)$ would adjoin an existing edge from $(x, y+2)$ to $(x+1, y+2)$. A side may also adjoin an edge in a switch to which it is not parallel. For example, the side from $(x, y+1)$ to $(x+1, y+1)$ could adjoin an existing edge from $(x, y+2)$ to $(x+1, y+2+1/3)$. Finally, exactly one of the four sides will be designated as the *docking side*. The docking side must adjoin either a side of a previously added buffer or a permissible edge of the gadget expansion.

When adding a buffer, if its sides adjoin existing sides or edges, then we also add edges connecting the buffer to other vertices and possibly also subdivide the adjoined edges or sides. There are four cases, depending on whether the newly added side is docking or non-docking and whether it adjoins a side of another buffer or an edge of the gadget expansion. These four cases are illustrated in Figure 12. In particular,

- the docking side of the added buffer is always subdivided four times and has four edges connecting it to the side or edge it adjoins (see Figure 12(a) and (b));
- a non-docking side adjoining an edge of the gadget expansion is subdivided once and has two connecting edges (see Figure 12(c));
- a non-docking side adjoining another buffer's side is subdivided thrice and has three connecting edges (see Figure 12(d));
- a (non-docking) side which does not adjoin anything is subdivided once.

Figure 12 explains the positions of the sides' subdivisions by way of an example for the right-side case. For the case of the other three sides, the coordinates are obtained by rotating around $(x+1/2, y+1/2)$.

The docking side must adjoin another buffer's side or a permissible edge. We will now discuss a strategy to achieve this. First, observe that every position $(x, y)$ at which we intend to add a buffer lies in a face of $(G', \mathrm{emb})$ that borders more than four vertices (possibly the unbounded face). In order to distinguish between such faces and faces within ladder paths, we will refer to the former as *regions*. We will add the buffers region-by-region. Next, note that once a buffer has been added to a region, then any subsequent buffer in that region can have its docking side adjoined to a (non-docking) side of another buffer added before it, as all edges on non-docking sides of a buffer are permissible by Lemma 5.7. (The gadget expansion is "surrounded" with buffers, so this works for the unbounded face.) Thus, it suffices to show how to add the first buffer for each region.

To this end, we must examine the structure of the gadget expansion. Let $R$ be any region. Clearly, $R$ borders some vertex gadget, and thus also a ladder path $L$ attached to an outside outlet of that vertex gadget. More precisely, $R$ borders either every inside or every outside edge of $L$. By construction of the gadget expansion, the first half of $L$ is attached to an outside outlet of some vertex gadget. Thus, we can find two permissible edges of $L$ that border $R$ by Lemma 5.8. Because we have two permissible edges to choose from, we can ensure that we never adjoin the docking sides of two buffers to two "parallel" edges of $L$ (i.e., to $\{u_i^j, u_{i+1}^j\}$ and $\{\tilde{u}_i^j, \tilde{u}_{i+1}^j\}$), as every ladder path only borders two regions. ⌟

We will use the following minor technical lemma.

▶ **Lemma 5.10 (★).** *Consider a square with corners $x_0, \ldots, x_3 \in \mathbb{R}^2$ and $P \subseteq \mathbb{R}^2$. Let $S_0, \ldots, S_3$ be the sides of the square, where $x_i$ is incident to $S_i$ and $S_{i+1}$ (all indices are modulo 4). Let $y_i := \frac{x_{i-1}+x_i}{2}$ for each $i \in \{1,2,3\}$ and $y_0 := \frac{x_3+x_0}{2}$. If $P$ contains $x_0, \ldots, x_3$ and possibly further points on the boundary of the square, but no points inside or outside of the square, then $\mathrm{RNG}(P) = \mathrm{GG}(P) = C_{|P|}$ if either of the following two conditions are met:*

**(i)** *$y_0, \ldots, y_3 \in P$ or*

**(ii)** *there is an $I \subseteq \{1,2,3,4\}$ such that $|I| \geq 3$ and $S_i \cap P \subseteq \{x_0, x_1, x_2, x_3\}$ for each $i \in I$.*

We are now prepared to prove the main result of this section.

**Proof of Theorem 5.1.** The proof builds on Construction 5.9. By Lemma 5.5, gadget expansion preserves Hamiltonicity. Each addition of a cycle involves subdividing a permissible edge (preserving Hamiltonicity by Lemma 5.6), and then adding a permissible cycle (preserving Hamiltonicity by Lemma 5.7). It follows that the construction preserves Hamiltonicity.

The construction already describes an embedding of the resulting graph $G'$. So it only remains to show that this embedding induces $G'$ as its RNG and GG. Let

$$\mathcal{A} := \{(x,y) \in \mathbb{Z}^2 \mid 14 \leq x \leq 16n+11,\ -6h_2(G) - 2 \leq y \leq 6h_1(G) + 3\}.$$

Note that for most $(x,y) \in \mathcal{A}$ there is a vertex embedded at $(x,y)$. The only exceptions are positions surrounding switches. For any $(x,y) \in \mathcal{A}$ we will call the vertices embedded at $(x,y)$, $(x+1,y)$, $(x,y+1)$, and $(x+1,y+1)$ along with any vertices embedded on the line segments between those four points a *grid face*. There are three classes of grid faces: grid faces within ladder paths or vertex gadgets, buffers, and grid faces between the aforementioned ones.

Within ladder paths, only two grid faces, which are shown in Figure 13 ($A$ and $B$), can occur. For these, the claim is implied by Lemma 5.10(ii). Within buffers, more variations are possible (e.g. $C$ and $D$ in Figure 13). Here, the claim is implied by Lemma 5.10(i). All grid faces between cycles or between cycles and vertex gadgets are pictured in Figure 12. It is easy to see that all pairs of non-adjacent vertices have GG blockers and no pair of adjacent vertices has an RNG blocker.

We now consider the area surrounding a switch. This area is pictured in Figure 13. The vertex $e$ is not a GG blocker for $\{a,b\}$, because $\mathrm{d}(a,b)^2 = 1/9$ and $\mathrm{d}(a,e)^2 = \mathrm{d}(b,e)^2 = 5/18$. The vertex marked $d$ is also not a GG blocker for $\{b,c\}$, since $\mathrm{d}(b,c)^2 = 16/9$, $\mathrm{d}(b,d)^2 = 10/9$, and $\mathrm{d}(c,d)^2 = 2$. Other cases are analogous or easy to see. E.g., $c$ and $e$ are not adjacent, because $b$ is a blocker. Vertices that do not share a grid face are not adjacent by Lemma 2.1.

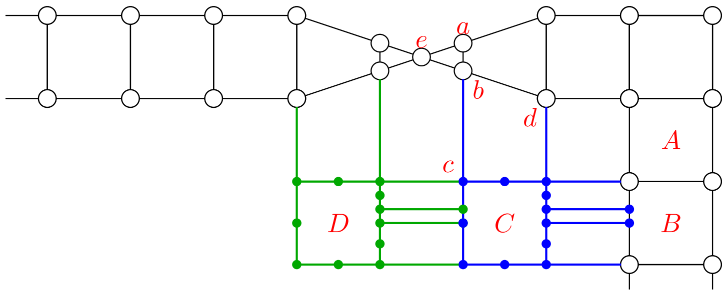


**Figure 13** An excerpt of the graph $G'$ produced by the reduction: Grid faces in $G'$: (A) Grid face within a ladder path or a vertex gadget with no docking side adjoined to its edges. (B) Grid face within a ladder path or a vertex gadget with a docking side adjoined to one of its edges. (C) + (D) Grid faces within buffers with the docking side on the right.

If $n$ is the number of vertices in the input graph $G$, then the graph $G'$ output by the construction contains $n$ vertex gadgets each containing $\mathcal{O}(1)$ vertices, $\mathcal{O}(n)$ ladder paths with $\mathcal{O}(n)$ vertices, and $\mathcal{O}(n^2)$ cycles with $\mathcal{O}(1)$ vertices. It is easy to see that each step in both constructions can be computed in polynomial time. Along with Proposition 5.2, this implies that HC is NP-hard on RNGs and on GGs. Moreover, it also implies that HC cannot be decided by a $2^{o(n^{1/4})}$-time algorithm on RNGs or GGs, unless the ETH fails. ◀

The computational complexity of Hamiltonian Cycle on RCGs and that of HC on RNGs and GGs with maximum degree 3 is left open.

## 6 Dominating Set and Independent Set

In a graph $G = (V, E)$, an *independent set* is a vertex set $X \subseteq V$ such that $G[X]$ is edgeless. In the Independent Set problem (IS), one is given a graph and $k \in \mathbb{N}_0$ and asked to decide whether the graph contains an independent set of size at least $k$. A *dominating set* in $G = (V, E)$ is a vertex set $X \subseteq V$ such that $N_G[X] = V$. In the Dominating Set problem (DS), one is given a graph and $k \in \mathbb{N}_0$ and asked to decide whether the graph contains a dominating set of size at most $k$. We also studied these two problems and proved:

▶ **Theorem 6.1 (★).** Dominating Set *and* Independent Set *on RCGs, on RNGs, and on GGs are* NP*-hard, even if the maximum degree is* 4*, Moreover, unless the ETH fails, neither problem admits a* $2^{o(n^{1/4})}$*-time algorithm where* $n$ *is the number of vertices.*

The proof, which is based on reductions from the corresponding problems on planar graphs with maximum degree 3, is deferred to the full version. It remains open whether DS or IS can be solved in polynomial time when restricted to RCGs, RNGs, or GGs with maximum degree 3.

## 7 Conclusion

We have shown that problems that are NP-hard on planar graphs typically remain NP-hard on the three proximity graph classes we study. This suggests that the main tools of algorithm theory to attack these problems shall be parameterized and approximation algorithms. IS, DS, and FVS all admit polynomial-time approximation schemes on arbitrary planar graphs, [2, 25] including the three types of proximity graphs we considered. FVS is fixed-parameter tractable on arbitrary graphs [5], while DS and IS are on planar graphs [1, 16]. We are not aware of any improvements (in terms of running time or approximation guarantees) to these results that are specific to RCGs, RNGs, or GGs.

It remains an important open question whether or not RCGs, RNGs, or GGs can be recognized in polynomial time and whether an embedding for a given graph can be computed in polynomial time [7, 8, 17]. If not, then one might suspect that the graph problems we have investigated might be easier if one is given an embedding rather than just the graph. Our reductions, however, prove that this is not the case, since we also give embeddings for the output graphs, which can easily be computed along with those output graphs.

We showed that FVS is NP-hard on proximity graphs with maximum degree 4, and it was already known to be polynomial-time solvable on any graph with maximum degree 3. For the other problems, we did not prove tight bounds on the maximum degree and this remains open. We proved ETH-based lower bounds of $2^{o(n^{1/4})}$ for each of these problems on RCGs, RNGs, and GGs (with the exceptions of 3-Col and HC on RCGs). On planar graphs in general, these problems can be solved in time $2^{\mathcal{O}(n^{1/2})}$ and this running time is optimal unless the ETH fails [28]. However, it might be possible to solve these problems on RCGs, RNGs, and GGs with a time bound strictly between $2^{o(n^{1/4})}$ and $2^{\mathcal{O}(n^{1/2})}$.

More generally, we are not aware of any problem that is known to be easier on the three graph classes we studied than on arbitrary planar graphs (excluding trivial cases like 3-Col on RCGs). Any such example would be of interest.

We conclude by remarking that most of the studied problems also remain hard on another canonical class of plane graphs, *Delaunay triangulations*. As the proofs work differently, we defer the specifics to subsequent publications.

## References


1 Jochen Alber, Hans L. Bodlaender, Henning Fernau, Ton Kloks, and Rolf Niedermeier. Fixed parameter algorithms for dominating set and related problems on planar graphs. *Algorithmica*, 33(4):461–493, 2002. doi:10.1007/s00453-001-0116-5.

2 Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *Journal of the ACM*, 41(1):153–180, January 1994. doi:10.1145/174644.174650.

3 Michael A. Bekos, Martin Gronemann, and Chrysanthi N. Raftopoulou. Two-page book embeddings of 4-planar graphs. *Algorithmica*, 75(1):158–185, 2016. doi:10.1007/s00453-015-0016-8.

4 Mark de Berg, Otfried Cheong, Marc van Kreveld, and Mark Overmars. *Computational Geometry*. Springer, 2008. doi:10.1007/978-3-540-77974-2.

5 Hans L. Bodlaender. On disjoint cycles. *International Journal of Foundations of Computer Science*, 5(1):59–68, 1994. doi:10.1142/S0129054194000049.

6 Prosenjit Bose, Vida Dujmović, Ferran Hurtado, John Iacono, Stefan Langerman, Henk Meijer, Vera Sacristán, Maria Saumell, and David R Wood. Proximity graphs: $E$, $\delta$, $\Delta$, $\chi$ and $\omega$. *International Journal of Computational Geometry & Applications*, 22(05):439–469, 2012. doi:10.1142/S0218195912500112.

7 Prosenjit Bose, William Lenhart, and Giuseppe Liotta. Characterizing proximity trees. *Algorithmica*, 16(1):83–110, 1996. doi:10.1007/BF02086609.

8 Franz Brandenburg, David Eppstein, Michael T. Goodrich, Stephen Kobourov, Giuseppe Liotta, and Petra Mutzel. Selected open problems in graph drawing. In *Proceedings of the 11th International Symposium on Graph Drawing (GD)*, pages 515–539, 2004. doi:10.1007/978-3-540-24595-7_55.

9 Rowland Leonard Brooks. On colouring the nodes of a network. *Mathematical Proceedings of the Cambridge Philosophical Society*, 37(2):194–197, 1941. doi:10.1017/S030500410002168X.

10 Jean Cardinal, Sébastien Collette, and Stefan Langerman. Empty region graphs. *Computational Geometry*, 42(3):183–195, 2009. doi:10.1016/J.COMGEO.2008.09.003.

11 Robert Cimikowski. Coloring proximity graphs. In *Memoranda in Computer and Cognitive Science: Proceedings of the First Workshop on Proximity Graphs*, pages 141–156, 1989. doi:10.21236/ada237250.

12 Robert Cimikowski. Coloring certain proximity graphs. *Computers & Mathematics with Applications*, 20(3):69–82, 1990. doi:10.1016/0898-1221(90)90032-F.

13 Robert J Cimikowski. Properties of some Euclidean proximity graphs. *Pattern Recognition Letters*, 13(6):417–423, 1992. doi:10.1016/0167-8655(92)90048-5.

14 Reinhard Diestel. *Graph Theory*. Springer, 5th edition, 2016. doi:10.1007/978-3-662-53622-3.

15 Michael B. Dillencourt. Finding Hamiltonian cycles in Delaunay triangulations is NP-complete. *Discrete Applied Mathematics*, 64(3):207–217, 1996. doi:10.1016/0166-218X(94)00125-W.

16 Rodney G. Downey and Michael R. Fellows. *Fundamentals of Parameterized Complexity*. Springer, 2013. doi:10.1007/978-1-4471-5559-1.

17 Peter Eades and Sue Whitesides. Nearest neighbour graph realizability is NP-hard. In *Proceedings of the 2nd Latin American Symposium on Theoretical Informatics (LATIN)*, pages 245–256, 1995. doi:10.1007/3-540-59175-3_93.

18 K. Ruben Gabriel and Robert R. Sokal. A new statistical approach to geographic variation analysis. *Systematic Biology*, 18(3):259–278, 1969. doi:10.2307/2412323.

19 M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976. doi:10.1016/0304-3975(76)90059-1.

20 M. R. Garey, D. S. Johnson, and R. Tarjan. The planar Hamiltonian circuit problem is NP-complete. *SIAM Journal on Computing*, 5(4):704–714, 1976. doi:10.1137/0205049.

21 Branko Grünbaum. Grötzsch's theorem on 3-colorings. *Michigan Mathematical Journal*, 10(3):303–310, 1963. doi:10.1307/mmj/1028998916.

22 Russell Impagliazzo and Ramamohan Paturi. On the complexity of $k$-SAT. *Journal of Computer and System Sciences*, 62(2):367–375, 2001. doi:10.1006/jcss.2000.1727.

23 J. W. Jaromczyk and M. Kowaluk. A note on relative neighborhood graphs. In *Proceedings of the 3rd Annual Symposium on Computational Geometry (SoCG)*, pages 233–241, 1987. doi:10.1145/41958.41983.

24 J. W. Jaromczyk and G. T. Toussaint. Relative neighborhood graphs and their relatives. *Proceedings of the IEEE*, 80(9):1502–1517, 1992. doi:10.1109/5.163414.

25 Jon Kleinberg and Amit Kumar. Wavelength conversion in optical networks. *Journal of Algorithms*, 38(1):25–50, 2001. doi:10.1006/jagm.2000.1137.

26 Pascal Kunz, Till Fluschnik, Rolf Niedermeier, and Malte Renken. Most classic problems remain NP-hard on relative neighborhood graphs and their relatives, 2021. doi:10.48550/ARXIV.2107.04321.

27 Philip M Lankford. Regionalization: Theory and alternative algorithms. *Geographical Analysis*, 1(2):196–212, 1969. doi:10.1111/j.1538-4632.1969.tb00615.x.

28 Daniel Lokshtanov, Dániel Marx, and Saket Saurabh. Lower bounds based on the exponential time hypothesis. *Bulletin of EATCS*, 105:41–71, 2011. URL: http://bulletin.eatcs.org/index.php/beatcs/article/view/92.

29 László Lovász. Three short proofs in graph theory. *Journal of Combinatorial Theory Series B*, 19(3):269–271, 1975. doi:10.1016/0095-8956(75)90089-1.

30 David W. Matula and Robert R. Sokal. Properties of Gabriel graphs relevant to geographic variation research and the clustering of points in the plane. *Geographical Analysis*, 12(3):205–222, 1980. doi:10.1111/j.1538-4632.1980.tb00031.x.

31 Joseph SB Mitchell and Wolfgang Mulzer. Proximity algorithms. In *Handbook of Discrete and Computational Geometry*, chapter 32, pages 849–874. Chapman and Hall/CRC, 3rd edition, 2017.

**32** Neil Robertson, Daniel Sanders, Paul Seymour, and Robin Thomas. Efficiently four-coloring planar graphs. In *Proceedings of the 28th Annual ACM Symposium on Theory of Computing (STOC)*, pages 571–575, 1996. doi:10.1145/237814.238005.

**33** Ewald Speckenmeyer. *Untersuchungen zum Feedback Vertex Set Problem in ungerichteten Graphen* [Investigations into the Feedback Vertex Set problem in Undirected Graphs]. PhD thesis, Universität Paderborn, 1983.

**34** Godfried T. Toussaint. The relative neighbourhood graph of a finite planar set. *Pattern Recognition*, 12(4):261–268, 1980. doi:10.1016/0031-3203(80)90066-7.

**35** Godfried T. Toussaint. Applications of the relative neighbourhood graph. *International Journal of Advances in Computer Science and Its Applications*, 4(3):77–85, 2014. URL: http://journals.theired.org/journals/paper/details/4323.html.

**36** Shuichi Ueno, Yoji Kajitani, and Shin'ya Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1-3):355–360, 1988. doi:10.1016/0012-365X(88)90226-9.

**37** Roderick B. Urquhart. Some properties of the planar Euclidean relative neighbourhood graph. *Pattern Recognition Letters*, 1(5):317–322, 1983. doi:10.1016/0167-8655(83)90070-3.