Planar 3-SAT with a Clause/Variable Cycle

Alexander Pilz[1]
Department of Computer Science, ETH Zürich. Zürich, Switzerland
alexander.pilz@inf.ethz.ch
https://orcid.org/0000-0002-6059-1821

Abstract

In the Planar 3-SAT problem, we are given a 3-SAT formula together with its incidence graph, which is planar, and are asked whether this formula is satisfiable. Since Lichtenstein's proof that this problem is NP-complete, it has been used as a starting point for a large number of reductions. In the course of this research, different restrictions on the incidence graph of the formula have been devised, for which the problem also remains hard.

In this paper, we investigate the restriction in which we require that the incidence graph is augmented by the edges of a Hamiltonian cycle that first passes through all variables and then through all clauses, in a way that the resulting graph is still planar. We show that the problem of deciding satisfiability of a 3-SAT formula remains NP-complete even if the incidence graph is restricted in that way and the Hamiltonian cycle is given. This complements previous results demanding cycles only through either the variables or clauses.

The problem remains hard for monotone formulas and instances with exactly three distinct variables per clause. In the course of this investigation, we show that monotone instances of Planar 3-SAT with three distinct variables per clause are always satisfiable, thus settling the question by Darmann, Döcker, and Dorn on the complexity of this problem variant in a surprising way.

2012 ACM Subject Classification Theory of computation → Problems, reductions and completeness

Keywords and phrases 3-SAT, 1-in-3-SAT, planar graph

Digital Object Identifier 10.4230/LIPIcs.SWAT.2018.31

Related Version `https://arxiv.org/abs/1710.07476`

Acknowledgements The Linked Planar 3-SAT problem has been brought to our attention by Lena Schlipf. We thank Erik Demaine for pointing out the motivation in platform games, Patrick Schnider and Michael Hoffmann for valuable discussions, as well as an anonymous reviewer for the indicated simplification.

1 Introduction

Let ϕ be a Boolean formula in conjunctive normal form (CNF) and let G_ϕ be a graph whose vertices are the variables and the clauses of ϕ such that (1) every edge of G_ϕ is between a variable and a clause and (2) there is an edge between a variable v and a clause c if and only if v occurs in c (negated or unnegated). We call ϕ a *CNF formula* and G_ϕ is called the *incidence graph* of ϕ. A CNF formula is a *3-SAT formula* if every clause contains at most three variables. (We will also discuss the case where every clause contains exactly three distinct variables.) The Planar 3-SAT problem asks whether a given 3-SAT formula ϕ is

[1] Supported by a Schrödinger fellowship of the Austrian Science Fund (FWF): J-3847-N35.

16th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2018).
Editor: David Eppstein; Article No. 31; pp. 31:1–31:13
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

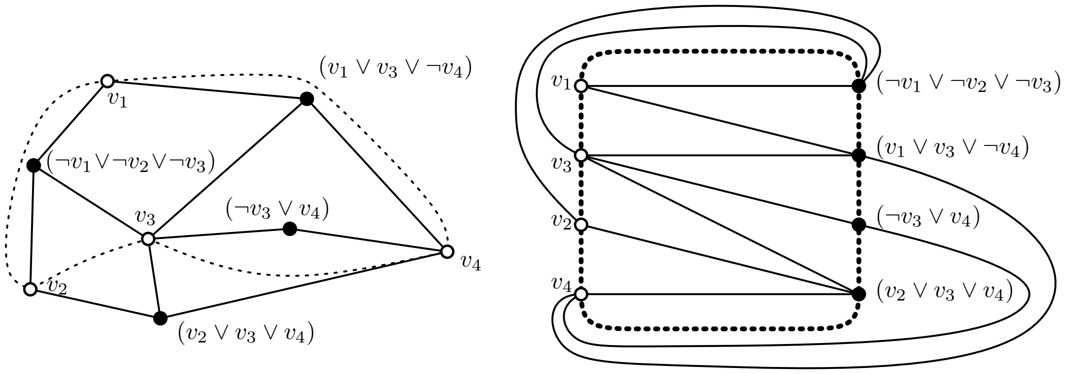
$(v_1 \vee v_3 \vee \neg v_4)$
v_1
$(\neg v_1 \vee \neg v_2 \vee \neg v_3)$
v_3
$(\neg v_3 \vee v_4)$
v_4
v_2
$(v_2 \vee v_3 \vee v_4)$
v_1
$(\neg v_1 \vee \neg v_2 \vee \neg v_3)$
v_3
$(v_1 \vee v_3 \vee \neg v_4)$
v_2
$(\neg v_3 \vee v_4)$
v_4
$(v_2 \vee v_3 \vee v_4)$

Figure 1 Left: An incidence graph for a 3-SAT formula that is planar. Clauses are black vertices and variables are white vertices. The graph is augmented by a spanning cycle (dotted) on the variable vertices. Right: A Linked Planar 3-SAT instance, with a Hamiltonian cycle κ (dotted) passing through the variables and the clauses. (At most two edges of κ may already be present in the incidence graph.)

satisfiable, given that G_ϕ is a planar graph. This problem has been shown to be NP-complete by Lichtenstein [21]. (In contrast to the general version, a PTAS is known for maximizing the number of satisfied clauses for the planar version of the 3-SAT problem [16].) See Figure 1 for drawings of an incidence graph.

Reducing from Planar 3-SAT is a standard technique to show NP-hardness of problems in computational geometry. In these reductions, the vertices and edges of G_ϕ are replaced by gadgets (consisting of geometric objects) that influence each other. However, it is often useful to have further restrictions on G_ϕ or on how G_ϕ can be embedded.

Lichtenstein's reduction already contains such a restriction: the problem remains NP-complete even if the graph remains planar after adding a cycle whose vertices are exactly the variables of the formula [21], and this cycle is part of the input. We call it a *variable cycle*. This fact allows for placing the variable vertices along a line, connected to "three-legged" clauses above and below that line (stated more explicitly in [18]; see Figure 2). De Berg and Khosravi [7] showed that it is also possible to have all literals in the clauses above the line to be positive, and all literals in clauses below the line negative. In Lichtenstein's reduction, one may as well add a *clause cycle* whose vertices are exactly the clauses of the formula while keeping the graph planar [20]. We call the Planar 3-SAT variants in which we require a variable cycle and a clause cycle Var-Linked Planar 3-SAT and Clause-Linked Planar 3-SAT, respectively.[2] Even though the incidence graphs constructed by Lichtenstein can be augmented with both a clause cycle [20] and a variable cycle, one cannot adapt Lichtenstein's construction to always obtain both cycles without any crossings.[3]

Our research is motivated by the following problem that attempts to combine these restrictions. See Figure 1 (right) for an accompanying illustration.

▶ **Definition 1** (Linked Planar 3-SAT)**.** Let $G_\phi = (C \cup V, E)$ be the incidence graph of a 3-SAT formula ϕ, where C is the set of clauses and V is the set of variables of ϕ. Further,

[2] While Lichtenstein's definition of planar 3-SAT [21] already requires the cycle through the variables, this property is often considered an explicit restriction. We thus follow the terminology of Fellows et al. [11] to emphasize when the variable cycle is needed.

[3] If not stated otherwise, we will implicitly require the incidence graphs augmented by the additional edges to be planar throughout this paper.

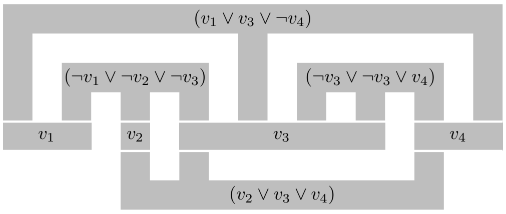
$(v_1 \vee v_3 \vee \neg v_4)$
$(\neg v_1 \vee \neg v_2 \vee \neg v_3)$
$(\neg v_3 \vee \neg v_3 \vee v_4)$
v_1
v_2
v_3
v_4
$(v_2 \vee v_3 \vee v_4)$

Figure 2 A "three-legged" Planar 3-SAT instance with variables on a line similar to [18, p. 425]. (There, two-variable clauses are transformed to three-variable clauses that contain one literal twice, a construction that is not necessary but possible for the initial graph in our reduction.)

let κ be a Hamiltonian cycle of $C \cup V$ that first visits all elements of C and then all elements of V. Suppose that the union of G_ϕ and κ is a planar graph. The Linked Planar 3-SAT problem asks, given ϕ, G_ϕ, and κ, whether ϕ is satisfiable.

Related problems in which all variables or all clauses can be drawn incident to the unbounded face are known to be in P, due to results by Knuth [17], and Kratochvíl and Křivánek [19], respectively. In particular, this is the case when there is a variable cycle and a path connecting all clauses or vice versa. One way of tackling the Linked Planar 3-SAT problem could be to show that G_ϕ has bounded treewidth. For such instances, the satisfiability of ϕ can be decided in polynomial time [12]. (This generalizes the above-mentioned results, as every k-outerplanar graph has treewidth at most $3k - 1$ [4]; see also Demaine's lecture notes [10].) However, with the right perspective on the Linked Planar 3-SAT problem, it will be easy to observe that there are formulas whose incidence graph has a grid minor with a linear number of vertices (and thus such graphs have unbounded treewidth). It is the same perspective through which we will show NP-completeness of the problem in Section 2, using a reduction from Planar 3-SAT. We note that requiring an arbitrary Hamiltonian cycle is not a restriction: As the incidence graph is bipartite, it is known that its page number is two [8]; hence, we can always add a Hamiltonian cycle through the variables and clauses in a planar way (possibly re-using edges of the incidence graph).

1.1 Motivation

Restrictions on the problem to reduce from can make NP-hardness reductions simpler. For reductions from Planar 3-SAT, it is common to actually reduce from Var-Linked Planar 3-SAT, using the variable cycle, in particular the "three-legged" embedding of [18]. Also, the clause cycle has been used [20, 11]. For an exhaustive survey on the numerous variants of Planar 3-SAT, see the thesis of Tippenhauer [27].

One motivation for considering Linked Planar 3-SAT is the framework for showing NP-hardness of platform games by Aloupis et al. [2]. In this class of reductions from 3-SAT to such games, a player's character starts at a specified position and traverses all variable gadgets, making a decision on their truth value. Clauses connected to the satisfied literal can be "unlocked" by visiting these clauses. Finally the player's character has to traverse all clause gadgets to reach the finish (called the "check path"). The framework then requires a game-specific implementation of the gadgets for start, finish, variables, clauses, and crossovers. Reducing from Linked Planar 3-SAT removes this dependency on crossover gadgets. (In particular, Theorem 9 can be used to show that the traversal through the literals can be done without crossings.) See [2, Section 2.1] for a more detailed description.

1.2 Results

We first prove that Linked Planar 3-SAT is NP-complete. In the following sections, we refine the construction to show that restricted variants of the problem remain hard as well. In particular, we do this for formulas without negated and unnegated variables in the same clause (Monotone Planar 3-SAT), and formulas for which the edges to negated variables are all on the same side of κ (recall that κ is the Hamiltonian cycle that first visits all variables and then all clauses). Also, we may require that all clauses contain exactly three distinct variables. A cycle κ in instances of Positive Planar 1-in-3-SAT (which requires exactly one true literal in each clause) also keeps the problem hard.

Finally, we discuss settings in which the planarity constraint is fulfilled only by satisfiable formulas. In particular, we show that planar CNF formulas with at least four variables per clause are always satisfiable. The same holds for instances of Monotone Planar 3-SAT with exactly three variables per clause. This solves an open problem by Darmann, Döcker, and Dorn [5, 6], who show that the corresponding problem with at most three variables per clause remains NP-complete with bounds on the variable occurrences, which refines the result of de Berg and Khosravi [7].

2 NP-hardness of Linked Planar 3-SAT

We now show that Linked Planar 3-SAT is NP-hard by reducing Planar 3-SAT to it. We are thus given a 3-SAT formula $\tilde{\phi}$ with variable set $\tilde{V}$, clause set $\tilde{C}$, and incidence graph $G_{\tilde{\phi}}$. Our goal is to construct another formula ϕ that is an instance of Linked Planar 3-SAT with incidence graph G_ϕ and Hamiltonian cycle κ such that ϕ is satisfiable if and only if $\tilde{\phi}$ is. The construction of G_ϕ will be given by an embedding.

We start by producing a suitable embedding of the initial graph $G_{\tilde{\phi}}$. We require an embedding Γ of $G_{\tilde{\phi}}$ on the integer grid, with edges drawn as x-monotone curves. (Our construction can easily be modified for non-x-monotone edges, but this assumption facilitates the presentation.) We further require that the size of the grid is polynomial in the size of $\tilde{\phi}$. It is well-known that a planar graph with n vertices can be embedded with straight-line edges on a $O(n) \times O(n)$ grid in $O(n)$ time [9, 26]. We can take such an embedding and perturb the clause vertices s.t. each variable has even x-coordinate, and the x-coordinate of each clause is odd. This can be done without introducing crossings by choosing the grid sufficiently large and scaling Γ; a blow-up by a factor polynomial in n is sufficient.[4] More specifically, we scale Γ by a polynomial multiple of 2 and increase the x-coordinate of each clause vertex by 1. (It will become apparent that our reduction merely uses the combinatorial structure of the embedding; however, it seems easier to describe the construction with a fixed straight-line embedding of $G_{\tilde{\phi}}$ on the integer grid.)

With a suitable drawing of $G_{\tilde{\phi}}$ at hand, we start the drawing of G_ϕ with the curve that will contain the cycle κ (we add its vertices later). It can be partitioned into two paths, one that will contain the elements of the variable set V (κ_V) and one for the elements of the clause set (κ_C). In our drawing shown in Figure 3 (left), we obtain a rectangular region R, whose intersection with κ consists of vertical line segments of unit distance, in alternation belonging to κ_V and κ_C. We call them the *clause segments* and *variable segments*, respectively. We

[4] We can consider the smallest horizontal distance between a vertex and a non-incident edge. This distance v is rational with numerator and denominator quadratic in the largest coordinate. By multiplying the x-coordinates of the vertices by $2/v$, rounding, and again multiplying by 2, we get the desired embedding, similar to [1, Lemma 6.1].

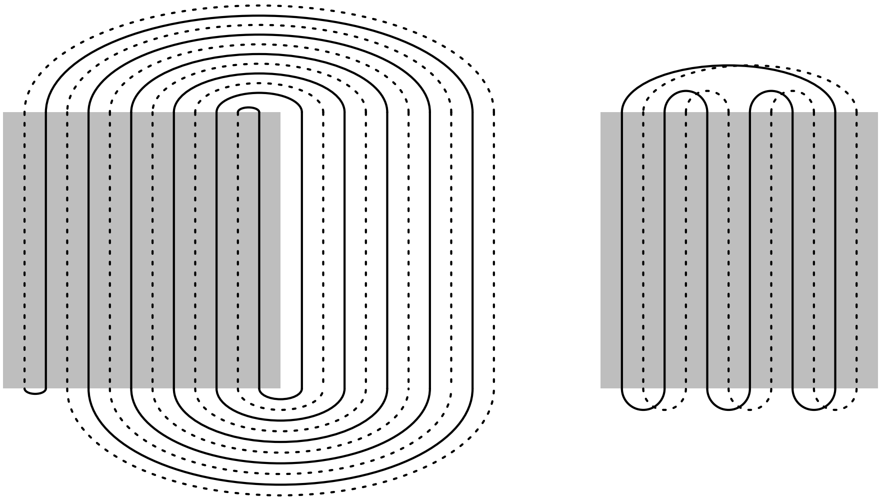

Figure 3 Left: Drawings of the paths κ_C (solid) and κ_V (dotted), containing the clauses and variables of ϕ, respectively. They intersect a rectangle R (gray) in vertical segments. The incidence graph is drawn inside R. Right: A similar construction can be used to show that all the incidence graphs we obtain can be augmented by either a clause cycle or a variable cycle (and not just two paths obtained from κ).

assume that the segments are placed on the integer grid with unit distance, with the variable segments having even x-coordinates and the clause segments having odd x-coordinates. In other words, R represents a grid in which the columns are traversed in alternation by κ_V and κ_C.

Place Γ inside R. By the construction of γ, all its variable vertices have even x-coordinates and are thus on variable segments, and its clause vertices are on clause segments. We obtain G_ϕ by replacing the edges of Γ by gadgets, consisting of subgraphs of G_ϕ. In the construction, we will make use of "cyclic implications", effectively copying the value of a variable; the graph G_ϕ will contain many pairs of variables x and x' with a clause $c_x = (\neg x \vee x')$ and a clause $c'_x = (x \vee \neg x')$. Clearly, $x = x'$ in any satisfying assignment. We depict the negation in such a clause c_x by an arrow from x to c_x, and from c_x to x' (where the arrows are also edges of the incidence graph). In general, we use the convention that an arrow from a variable to a clause denotes that the variable occurs negated in that clause, while an arrow from the clause to the variable means that the variable occurs unnegated.

We replace each edge e of Γ by a sequence of so-called *connector gadgets*. A connector gadget consists of two variables x and x', and two clauses $c_x = (\neg x \vee x')$ and $c'_x = (\neg x' \vee x)$, implying $x = x'$ in any satisfying truth assignment. The variable vertices are placed on the intersections of e with two consecutive variable segments in R, and the clause vertices are placed on the clause segment between them (also close to the crossing of e and the clause segment), effectively subdividing κ. An edge in Γ connecting a variable $\tilde{v} \in \tilde{V}$ to a clause $\tilde{c} \in \tilde{C}$ that crosses κ can be replaced by a sequence of connector gadgets in a sufficiently small neighborhood of the edge, as shown in Figure 4. Thus, in the resulting drawing, we have subdivided κ to remove crossings with e, and the resulting formula is satisfiable if and only if the initial formula is satisfiable.

Replacing the edges of Γ by the connector gadgets results in a drawing of G_ϕ. As all edges are x-monotone, all crossings with κ are replaced. Thus this drawing is planar and contains a number of vertices that is polynomial in $|C|$, as the number of crossings of an edge

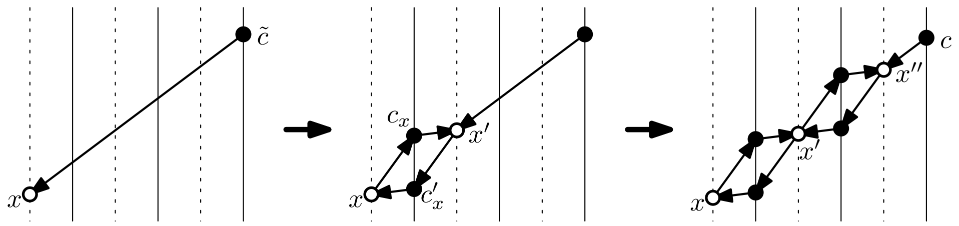

Figure 4 An edge of the initial incidence graph crossing the cycle κ (left) can be replaced by a sequence of connector gadgets. Variable vertices are white dots, clause vertices are black. The two variables in the connector gadget (middle) have the same value, and the occurrence of a variable x in the clause $\tilde{c}$ is replaced by the last variable x'' to obtain an equivalent clause c.

in Γ with clause and variable segments (and thus the number of vertices needed to replace the edge) is bounded by the grid size. Also, all "new" clauses contain only two variables (we will see a modification of the reduction without this property). As none of the edges in the resulting embedding of G_ϕ crosses the initially drawn cycle for κ, G_ϕ can be augmented by κ along that cycle maintaining planarity. Finally, observe that ϕ is satisfiable if and only if $\tilde{\phi}$ is: the two variables of a connector gadget must have the same value, and the clauses not part of the connector gadget are the clauses of $\tilde{C}$ in which we replaced variables by others that have to be equal. We thus obtain our main result.

▶ **Theorem 2.** Linked Planar 3-SAT *is NP-complete.*

Observe that our construction also does not change the number of satisfiable assignments, i.e., the reduction is *parsimonious.* Since counting the number of satisfiable assignments to a planar 3-SAT formula is #P-complete [15], this also holds for Linked Planar 3-SAT.

3 Further variants

We use the main idea of drawing the cycle κ as shown in Figure 3 to obtain similar reductions for variants of the planar satisfiability problem. If the initial problem is known to be hard, we merely need to find an according connector gadget to replace the crossings of edges of $G_{\tilde{\phi}}$ and κ.

3.1 Positive Planar 1-in-3-SAT

It is easy to transform our reduction to be an instance of 1-in-3-SAT, where exactly one variable is true: we get the "main" clauses directly from an initial 1-in-3-SAT instance, and the connector gadgets are the same. We therefore impose two more requirements on the instance. First, all clauses should have three elements, and second, all literals should be positive.

▶ **Definition 3** (Positive Planar 1-in-3-SAT [24]). Given a formula ϕ in which each clause contains exactly three distinct unnegated literals, and an embedding of the incidence graph G_ϕ, the Positive Planar 1-in-3-SAT problem asks whether there exists a satisfying assignment of ϕ such that exactly one variable in each clause is true.

Mulzer and Rote [24] show that deciding satisfiability of Positive Planar 1-in-3-SAT instances is hard, even if the incidence graph can be augmented by a variable cycle.

▶ **Theorem 4.** *The* Positive Planar 1-in-3-SAT *problem remains NP-complete even for problem instances that are also instances of the* Linked Planar 3-SAT *problem.*

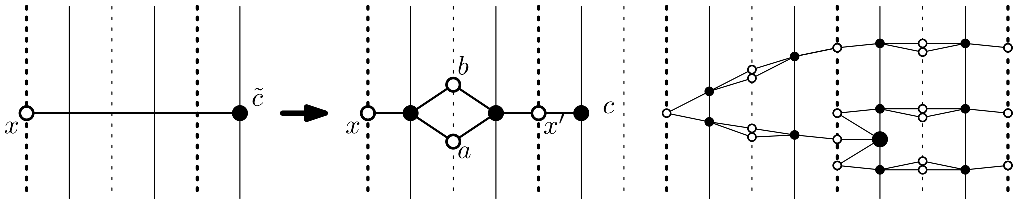

Figure 5 Reduction for the Positive Planar 1-in-3-SAT variant. If $x = 1$, then both $a = 0$ and $b = 0$, and thus $x' = 1$ as well. Otherwise, one of a and b is 1 and we again have $x = x'$. Since the connector gadgets have width 4, we may "overshoot" when coming from the right.

Proof. The reduction works again by blowing up the grid embedding of the initial Positive Planar 1-in-3-SAT incidence graph by a polynomial factor f and increase the x-coordinate of each clause by 1. We choose f to be a multiple of 8, as (i) all variable vertices should have even x-coordinates (and are thus placed on the appropriate part of κ), (ii) the number of clause segments between a variable and a clause (before increasing its x-coordinate by 1) is a multiple of two (because of the gadget width described below), and (iii) there are at least three variable segments between each clause vertex of the initial instance and each of its variables, even after increasing the x-coordinate of each clause vertex by 1 (to have sufficient space for "detours" described below). As before, f is chosen such that moving the clauses does not produce any crossings in the straight-line embedding. Then, we use two clauses sharing two variables to produce a connector gadget, as shown in Figure 5. The two clauses (x, a, b) and (x', a, b) ensure that $x = x'$.

The construction works analogous to the simpler connector gadget of the previous section,[5] except for the following caveat. The connector gadgets have width 4, and this is the reason for requiring condition (ii): We placed each clause of the initial formula at positions with x-coordinate $8k + 1$, for some integer k. The sketch on the right of Figure 5 shows how to connect the clause to the connector gadgets. (Note the "detour" one chain of connector gadgets may have to take in the vicinity of the initial clause to connect to it if all three edges of the clause emanate to the right in the initial drawing; this does not interfere with other gadgets as f is a multiple of 8.) ◀

3.2 Exactly three distinct variables per clause

If we require the formula to have exactly three distinct variables in each clause, the reduction can also be modified accordingly. Mansfield [22] showed how to extend Lichtenstein's construction to obtain a planar 3-SAT formula with exactly three different variables per clause[6] by constructing such a formula with planar incidence graph and a variable that is false in every satisfying assignment.

▶ **Theorem 5.** *The* Linked Planar 3-SAT *problem remains NP-complete even if each clause contains exactly three different variables.*

Proof. We reduce from the non-linked version of the problem by drawing an incidence graph on our grid and replace edge parts by gadgets that transport the truth settings of a variable. To this end, we use a modified connector gadget shown in Figure 6, which assure that two

[5] We thank an anonymous referee for simplifying the previously-used gadget.
[6] This restriction has been re-discovered, e.g., in [15].

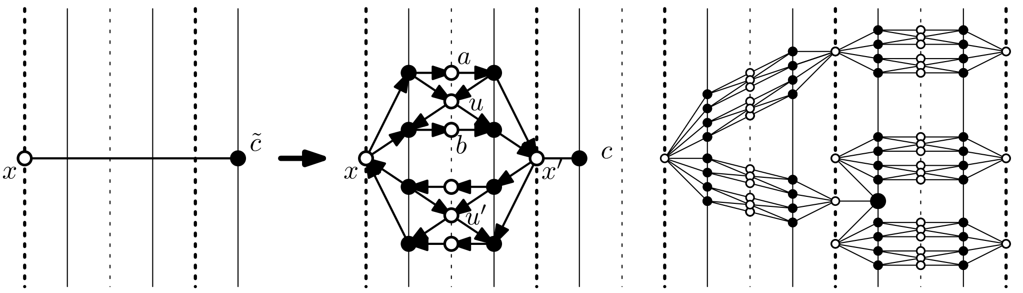

Figure 6 A modified connector gadget. The leftmost variable has to have the same value as the rightmost one. All clauses have exactly three distinct variables. An arrow from a variable to a clause indicates that the variable occurs negated. For an arrow from the clause to the variable, the occurrence is unnegated.

variables x and x' always have the same value: When removing u from the shown formula, we get $x \Rightarrow a$ and $a \Rightarrow x'$, as well as $x \Rightarrow b$ and $b \Rightarrow x'$. Since u occurs unnegated in the clauses containing a, and it occurs negated in the clauses containing b, one of the two implication pairs will make sure that $x \Rightarrow x'$. More formally, we have $(\neg x \vee a \vee u) \wedge (\neg a \vee u \vee x')$, which entails $(\neg x \vee u \vee x')$, and $(\neg x \vee b \vee \neg u) \wedge (\neg b \vee \neg u \vee x')$, which entails $(\neg x \vee \neg u \vee x')$. These two clauses have $(\neg x \vee x')$ as a resolvent. Analogously, the clauses containing u' imply $x' \Rightarrow x$.

We note that in Mansfield's reduction it is no longer shown that there exists a variable cycle, so we cannot rely on a "three-legged" embedding but rather use a straight-line one. As in the previous reduction, the connector gadgets have width 4. We thus again use a suitable blow-up factor and place each clause of the initial formula at positions with x-coordinate $8k + 1$, for some integer k. ◀

Note that the bi-implication implemented by the connector gadget has sixteen different truth assignments, independent of whether x is true or false. As the Planar 3-SAT problem is known to be #P-complete even if each clause contains exactly three distinct variables [15], we can add connector gadgets to transform any such instance into a Linked Planar 3-SAT instance. For each connector gadget, the number of solutions is multiplied by 16.

▶ **Theorem 6.** *The* Linked Planar 3-SAT *problem remains #P-complete even if each clause contains exactly three different variables.*

3.3 Monotonicity restrictions

We can add the following list of restrictions to the setting for which the problem remains hard. Both can be shown by reducing from Monotone Planar 3-SAT.

A clause is *monotone* if it contains either only negated or only unnegated literals; a formula is *monotone* if all its clauses are monotone.

▶ **Definition 7** (Monotone Planar 3-SAT)**.** Let $G_\phi = (C \cup V, E)$ be the incidence graph of a 3-SAT formula ϕ, where C is the set of clauses and V is the set of variables of ϕ, and each clause of C is monotone. The Monotone Planar 3-SAT problem asks whether ϕ is satisfiable.

This problem has been shown to be NP-complete by de Berg and Khosravi [7]; they actually show that the problem remains hard even if there is a variable cycle separating the clauses with the negated variables from the ones with the unnegated variables. That is,

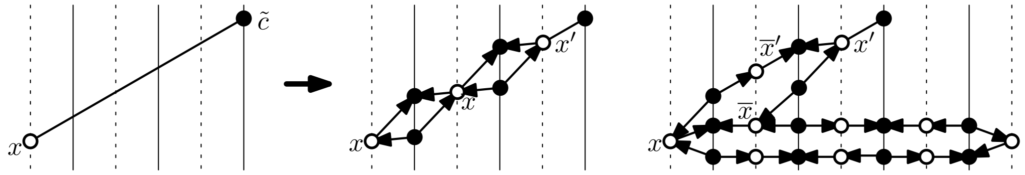

Figure 7 A variant of the connector gadget in which all clauses are monotone. We have $x \neq \overline{x} \neq x' = x$ in the middle. The gadget can be further split to have each variable vertex of degree at most three (right).

the incidence graph can be drawn in a rectilinear way, with the variables on the x-axis and exactly the clauses with the negated occurrences below the x-axis. We can use their result to show hardness of the according problem in our setting.

▶ **Theorem 8.** *The* Linked Planar 3-SAT *problem remains NP-complete even if all clauses are monotone.*

Proof. We reduce from Monotone Planar 3-SAT. The reduction is the same as for general Linked Planar 3-SAT, but with a different connector gadget. We use two sub-gadgets that contain two clauses, one all positive one all negative, that ensure that two variables are not equal. Two such gadgets in sequence form a new connector gadget. See Figure 7. We can deal with the position of the clause being off by one to the end of each connector gadget in the same way as in Figure 6 (right). ◀

Lichtenstein already showed that Planar 3-SAT remains NP-complete even if we require that the variable cycle partitions the edges of every variable vertex into those leading to a negated occurrence and those leading to an unnegated one [21, Lemma 1]. (As he mentions, this also implies that one could split a variable vertex into two literal vertices while preserving planarity.) Note that there is a subtle difference to the restriction of de Berg and Khosravi [7] to Monotone Planar 3-SAT, as the side of the variable cycle to which the edges to the negated occurrences emanate is not fixed globally. The following set of related restrictions could be particularly interesting for further reductions.

▶ **Theorem 9.** *The* Linked Planar 3-SAT *problem remains NP-complete even if, for each clause, the edges corresponding to positive occurrences emanate to the interior of the cycle κ, and the ones to negated occurrences to the exterior. In addition, each variable occurs in at most three clauses.*

Proof. We reduce from the Monotone Planar 3-SAT variant of de Berg and Khosravi [7], using a "three-legged" embedding. The construction (see Figure 8) uses cycles consisting of the variables $x_1, \ldots x_k$ and $\overline{x_1}, \ldots, \overline{x_k}$ and clauses $(\neg x_i \vee x_{i+1})$ and $(\neg\overline{x_i} \vee \overline{x_{i+1}})$ for all valid indices, as well as the clauses $(x_1 \vee \overline{x_1})$ and $(\neg x_k \vee \neg\overline{x_k})$. The latter thus entails $(\neg x_1 \vee \neg\overline{x_1})$. We therefore have $x_i = x_j$, $\overline{x_i} = \overline{x_j}$, and $x_i \neq \overline{x_j}$ for $i, j \leq k$. The variable vertices are placed from left to right with increasing indices. Thus, the variables on the upper part of the cycle have the opposite value of those on the lower part. For a clause with positive literals in the Monotone Planar 3-SAT instance, we connect the variables as in Figure 8.

We therefore have variable gadgets that consist of said cycles; see the three bottom-most cycles in Figure 8. These variable gadgets are connected to the clause c (which represents the clause of the initial formula) either directly (for the middle variable gadget), or via another cycle (for the left and the right variable gadget). For these connector gadgets, we again have that the variables on the upper part have the opposite value as those of the lower part; in the

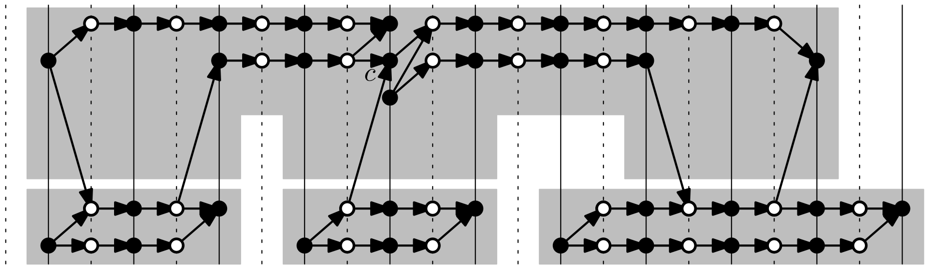

Figure 8 All clauses have negative literals to the left and positive literals to the right. The variable state is transported by cycles where the variables on top have the negative value of the variables at the bottom. A clause $\tilde{c} = (x \vee y \vee z)$ is transformed to $c = (\neg\overline{x_i} \vee \neg\overline{y_j} \vee z_k)$. The "three-legged" embedding of the clause is indicated by the gray contour.

way a variable gadget and a connector gadget are connected, a variable on the upper part of the variable gadget has the same value as one on the lower part of the connector gadget. The clause c has two edges to the left (which therefore correspond to negated occurrences) and one to the right (an unnegated occurrence). In the setting shown in Figure 8), we consider a variable set to true if the variables on the lower part of the variable gadget are true. A clause $\tilde{c} = (x \vee y \vee z)$ is thus transformed to $c = (\neg\overline{x_i} \vee \neg\overline{y_j} \vee z_k)$. We can have variable gadgets of arbitrary width simply by having larger cycles, and connect the clauses according to the "three-legged" embedding. The resulting construction is thus crossing-free, and the formula is satisfiable if and only if the initial one is. ◀

Observe that the connector gadgets in Figure 8 use only clauses with two variables. The reduction therefore also works for Planar 2-SAT instances. While Planar 2-SAT can be solved in polynomial time, it is known to be #P-complete [29]. As our gadgets are parsimonious (i.e., do not change the number of solutions), they can be applied to show #P-completeness.

▶ **Corollary 10.** *The* Linked Planar 2-SAT *problem is #P-complete. It remains #P-complete even if for each clause, the edges corresponding to positive occurrences emanate to the interior of the cycle κ, and the ones to negative literals to the exterior.*

Note that making each variable occur in at most three clauses requires that there are clauses with at most two (different) literals, as every CNF formula with exactly three literals per clause and at most three occurrences per variable is satisfiable [28].

Darmann, Döcker, and Dorn [5, 6] showed how to reduce the number of times a variable occurs. For the variant of Monotone Planar 3-SAT that requires exactly three different variables per clause, the complexity was previously unknown [5, 6]. Surprisingly, it turns out that such instances are always satisfiable, as discussed in the next section.

4 Remark: different cycles through clauses and variables

Observe that, for all our constructions, G_ϕ still allows for adding a variable cycle H, as well as a clause cycle H', as shown in Figure 3 (right). But these cycles will, in general, cross mutually. Also, they will cross the cycle κ. Recall that if H did not cross H' or κ_C, the problem would be solvable in polynomial time [10].

Using the gadgets and the two cycles shown in Figure 3 (right), we observe that Planar 3-SAT remains NP-complete even if these cycles exist for all variants mentioned. While the variable and clause cycles have been identified in [21] and [20], respectively, it seems to

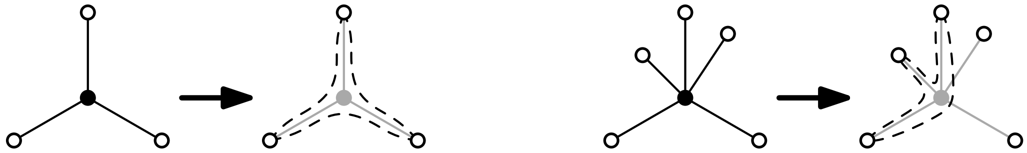

Figure 9 Clause (black) with at least three distinct variables (white). We can augment the graph with a 3-cycle through variables (dashed) and remove the clauses while preserving planarity. In any 4-coloring of the graph, the three variables belong to three different color classes and thus there always exists a satisfying truth assignment by setting the variables of two color classes to true.

have been unknown for the variants using exactly three variables per clause (even though Mansfield's construction [22] can be embedded to obtain the cycles). Also, it seems that clause cycles have not been considered for MONOTONE PLANAR 3-SAT and PLANAR POSITIVE 1-IN-3-SAT.

5 Properties forcing satisfiability

Planarity is a rather drastic combinatorial restriction on the structure of a graph. While NP-completeness of 3-SAT is preserved in the planar setting, further properties may lead not only to polynomial-time algorithms (as for PLANAR NAE-SAT [23]), but also to instances that are always satisfiable.

▶ **Theorem 11.** *Every instance of* PLANAR SAT *in which each clause has three negated or three unnegated occurrences of three distinct variables is satisfiable. A satisfying assignment can be found in quadratic time.*

Proof. Consider any plane drawing of the incidence graph of the formula. For each clause vertex, we can add a 3-cycle consisting of three of its variable vertices that are either all negated or all unnegated, as shown in Figure 9. (For clauses with three variables this is similar to a Y-Δ transform, a common operation to replace a vertex of degree 3 by a 3-cycle: connect two variables by a curve in a neighborhood of the two edges connecting them to the clause.) Observe that, after removing the clause vertices, the resulting graph is still planar. It is thus 4-colorable [3] and we may consider any 4-coloring of the variables. Set the variables of the vertices with colors 1 and 2 to true, and the others to false. A 3-cycle contains three different colors, and thus a 3-cycle through three (monotone) variable vertices has at least one variable set to true and one variable set to false. A 4-coloring can be found in quadratic time [25]. ◀

▶ **Corollary 12.** *Every instance of* MONOTONE PLANAR SAT *with at least three distinct variables per clause is satisfiable.*

▶ **Corollary 13.** *Every instance of* PLANAR SAT *with at least five distinct variables per clause is satisfiable.*

For at least four variables per clause, we can give a similar result, closing the gap between Corollary 13 and Mansfield's result [22], using less heavy machinery than the Four-Color theorem. The proof makes use of Hall's theorem [13], inspired by a technique by Tovey [28]: each subset of k clauses has at least k variables occurring in it.

▶ **Lemma 14.** *Let $G = (B \cup C, E)$ be a bipartite planar graph with parts B and C such that all vertices in C have degree at least four. Then there exists a matching covering every vertex of C.*

Proof. Consider any plane embedding of G and let F be its set of faces. Thus, by Euler's formula, we have $|B| + |C| - |E| + |F| = 1 + k$, where $k \geq 1$ is the number of connected components of G. Every edge has two incidences with faces, and since G is bipartite, every face has at least four incidences with edges. We get $2|E| = \sum_{f \in F} |f| \geq 4|F|$ (where $|f|$ is the number of sides of the face f). Combining this with Euler's formula to $|E| \geq 2(1 + k - |B| - |C| + |E|)$ gives the bound

$$2|B| + 2|C| - 2 - 2k \geq |E| \ . \tag{1}$$

Further, $|E| \geq 4|C|$, as G is bipartite and every element of C is incident to at least four edges. Combining this with (1) results in

$$|C| \leq |B| - 1 - k \ . \tag{2}$$

Finally, observe that every subset of C plus its neighbors in B also induce a graph in which the analogue of (2) holds. As, for every subset of C, the set of adjacent elements in B has at least the same cardinality, there is a matching on G that covers C by Hall's theorem [13]. ◀

▶ **Theorem 15.** *Each CNF formula with planar incidence graph of n vertices and at least four distinct variables per clause has a satisfying assignment, which can be found in $O(n^{1.5})$ time.*

Proof. Let ϕ be a planar SAT instance with vertex set V and clause set C, and let $G_\phi = (V \cup C, E)$ be the associated incidence graph. By Lemma 14, there is a matching on G_ϕ covering C, which assigns a distinct variable to each clause (i.e., a system of distinct representatives for the clauses). If we set the according literal to true, we get an assignment satisfying ϕ. The matching can be found using the algorithm by Hopcroft and Karp [14]. ◀

References

1 Oswin Aichholzer, Wolfgang Mulzer, and Alexander Pilz. Flip distance between triangulations of a simple polygon is np-complete. *Discrete & Computational Geometry*, 54(2):368–389, 2015. doi:10.1007/s00454-015-9709-7.

2 Greg Aloupis, Erik D. Demaine, Alan Guo, and Giovanni Viglietta. Classic nintendo games are (computationally) hard. *Theor. Comput. Sci.*, 586:135–160, 2015. doi:10.1016/j.tcs.2015.02.037.

3 Kenneth Appel and Wolfgang Haken. *Every Planar Map is Four-Colorable.* Number 98 in Contemporary Mathematics. AMS, 1989. With the collaboration of J. Koch.

4 Hans L. Bodlaender. A partial k-arboretum of graphs with bounded treewidth. *Theor. Comput. Sci.*, 209(1-2):1–45, 1998. doi:10.1016/S0304-3975(97)00228-4.

5 Andreas Darmann, Janosch Döcker, and Britta Dorn. On planar variants of the monotone satisfiability problem with bounded variable appearances. *CoRR*, abs/1604.05588, 2016. arXiv:1604.05588.

6 Andreas Darmann, Janosch Döcker, and Britta Dorn. The monotone satisfiability problem with bounded variable appearances. *Int. J. Foundations Comp. Sci.*, 2017. To appear.

7 Mark de Berg and Amirali Khosravi. Optimal binary space partitions for segments in the plane. *Int. J. Comput. Geometry Appl.*, 22(3):187–206, 2012. URL: http://www.worldscinet.com/doi/abs/10.1142/S0218195912500045.

8 Hubert de Fraysseix, Patrice Ossona de Mendez, and János Pach. A left-first search algorithm for planar graphs. *Discrete & Computational Geometry*, 13:459–468, 1995. doi:10.1007/BF02574056.

9 Hubert de Fraysseix, János Pach, and Richard Pollack. How to draw a planar graph on a grid. *Combinatorica*, 10(1):41–51, 1990. doi:10.1007/BF02122694.

10 Erik Demaine. Algorithmic lower bounds: Fun with hardness proofs; Lecture 7. http://courses.csail.mit.edu/6.890/fall14/scribe/lec7.pdf. Scribe: Quanquan Liu, Jeffrey Bosboom. Retrieved June 21, 2017.

11 Michael R. Fellows, Jan Kratochvíl, Matthias Middendorf, and Frank Pfeiffer. The complexity of induced minors and related problems. *Algorithmica*, 13(3):266–282, 1995. doi:10.1007/BF01190507.

12 Eldar Fischer, Johann A. Makowsky, and Elena V. Ravve. Counting truth assignments of formulas of bounded tree-width or clique-width. *Discrete Applied Mathematics*, 156(4):511–529, 2008. doi:10.1016/j.dam.2006.06.020.

13 Philip Hall. On representatives of subsets. *J. London Math. Soc.*, s1-10(1):26–30, 1935.

14 John E. Hopcroft and Richard M. Karp. An $n^{5/2}$ algorithm for maximum matchings in bipartite graphs. *SIAM J. Comput.*, 2(4):225–231, 1973. doi:10.1137/0202019.

15 Harry B. Hunt III, Madhav V. Marathe, Venkatesh Radhakrishnan, and Richard Edwin Stearns. The complexity of planar counting problems. *SIAM J. Comput.*, 27(4):1142–1167, 1998. doi:10.1137/S0097539793304601.

16 Sanjeev Khanna and Rajeev Motwani. Towards a syntactic characterization of PTAS. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 329–337. ACM, 1996. doi:10.1145/237814.237979.

17 Donald E. Knuth. Nested satisfiability. *Acta Inf.*, 28(1):1–6, 1990. doi:10.1007/BF02983372.

18 Donald E. Knuth and Arvind Raghunathan. The problem of compatible representatives. *SIAM J. Discret. Math.*, 5(3):422–427, 1992.

19 Jan Kratochvíl and Mirko Křivánek. Satisfiability of co-nested formulas. *Acta Inf.*, 30(4):397–403, 1993. doi:10.1007/BF01209713.

20 Jan Kratochvíl, Anna Lubiw, and Jaroslav Nesetril. Noncrossing subgraphs in topological layouts. *SIAM J. Discrete Math.*, 4(2):223–244, 1991. doi:10.1137/0404022.

21 David Lichtenstein. Planar formulae and their uses. *SIAM J. Comput.*, 11:329–343, 1982.

22 Anthony Mansfield. Determining the thickness of graphs is NP-hard. *Math. Proc. Cambridge Phil. Soc.*, 93(01):9–23, 1983.

23 Bernard M. E. Moret. Planar NAE3SAT is in P. *SIGACT News*, 19(2):51–54, 1988.

24 Wolfgang Mulzer and Günter Rote. Minimum-weight triangulation is np-hard. *J. ACM*, 55(2):11:1–11:29, 2008. doi:10.1145/1346330.1346336.

25 Neil Robertson, Daniel P. Sanders, Paul D. Seymour, and Robin Thomas. Efficiently four-coloring planar graphs. In Gary L. Miller, editor, *Proceedings of the Twenty-Eighth Annual ACM Symposium on the Theory of Computing, Philadelphia, Pennsylvania, USA, May 22-24, 1996*, pages 571–575. ACM, 1996. doi:10.1145/237814.238005.

26 Walter Schnyder. Embedding planar graphs on the grid. In David S. Johnson, editor, *Proceedings of the First Annual ACM-SIAM Symposium on Discrete Algorithms, 22-24 January 1990, San Francisco, California.*, pages 138–148. SIAM, 1990. URL: http://dl.acm.org/citation.cfm?id=320176.320191.

27 Simon Tippenhauer. On planar 3-SAT and its variants. Master's thesis, Freie Universität Berlin, 2016.

28 Craig A. Tovey. A simplified np-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85–89, 1984. doi:10.1016/0166-218X(84)90081-7.

29 Salil P. Vadhan. The complexity of counting in sparse, regular, and planar graphs. *SIAM J. Comput.*, 31(2):398–427, 2001. doi:10.1137/S0097539797321602.