Discrete Mathematics and Theoretical Computer Science DMTCS vol. **19**:3, 2017, #17

Total Domination, Connected Vertex Cover and Steiner Tree with Conflicts

Alexis Cornet Christian Laforest

LIMOS, France
Université Clermont Auvergne, France

received 9th June 2017, *revised 20th Sep. 2017*, *accepted 8th Dec. 2017*.

Total Dominating Set, Connected Vertex Cover and Steiner Tree are well-known graph problems. Despite the fact that they are NP-complete to optimize, it is easy (even trivial) to find some solutions, when ignoring the optimization criterion. In this paper, we study a variant of these problems by adding conflicts, that are pairs of vertices that cannot be both in a solution. This new constraint leads to situations where it is NP-complete to decide if there exists a solution avoiding conflicts. We prove NP-completeness of deciding the existence of a solution for different restricted classes of graphs and conflicts, improving recent results. We also propose polynomial time constructions in several restricted cases and we introduce a new parameter, the *stretch*, to capture the locality of the conflicts.

Keywords: conflicts, graph theory, forbidden pairs, connected vertex cover, total dominating set, Steiner tree, NP-completeness

1 Introduction

In the field of discrete optimization, graph problems have been extensively studied. An instance of a problem consists of a graph G (together with other parameters), and the goal is to compute a structure S which satisfies some constraints and whose value is optimized for a criterion. It is the case for domination problems, connected vertex cover, and Steiner tree for which coming up with approximation algorithms is a field of active research [ACG$^+$12]. However, deciding the existence of some solution ignoring the optimization criterion can easily be done in polynomial time.

In the real world, there can be incompatibilities between some vertices of G because of various reasons, for example structural incompatibilities between components of a system, security reasons, mutually exclusive funding, interface incompatibilities and so on. These plausible applications motivate us to extend several classical optimization problems to understand better how these incompatibilities reflect on the complexity of these problems. To model these situations, we say that two elements u and v are in *conflict* if u and v cannot be both in a solution. A graph $G = (V, E)$ will be called the *support graph*, and the set of conflicts will be interpreted as a graph on the same vertices, called the *conflict graph* C. No solution can contain both ends of an edge of C, i.e., any solution must be an independent set of C. A pair (G, C) where G is the support graph and C is the conflict graph will be called a *graph with conflicts*. In the following, we will not make distinction between a conflict between vertex a and vertex b and the edge

ISSN 1365–8050

ab of the conflict graph C. Given this additional constraint, deciding the existence of a solution becomes harder. In fact, we show that deciding the existence of a solution for Total Dominating Set with no conflicts, Connected Vertex Cover with no conflicts and Steiner Tree with no conflicts is NP-complete even for restricted graph classes.

Problems with conflicts, also known as forbidden pairs, have been studied in numerous problems. In a recent serie of papers [KLM13a, KLM13b, KMMN15, LM14, LM15], authors deal with conflicts between pairs of edges, in problems such as finding a path, spanning tree, Hamiltonian path and Hamiltonian cycle. Most of the results are NP-completeness theorems on the existence of such objects. Conflicts between vertices have also been studied until recently by several authors. Most of the works [GMO76, KP09, Kov13, Yin97] study the complexity of finding paths without conflicts. In [DLP16], the authors prove the NP-completeness of deciding the existence of a solution in some problems with conflicts, including domination problems, Connected Vertex Cover and Steiner Tree. However, their reductions use graphs of unbounded maximum degree, and a lot of conflicts. In an ongoing work [CL], we prove the NP-completeness of deciding the existence of a dominating set with no conflicts and independent dominating set with no conflicts in very restricted classes of graphs.

In this paper, in sections 2, 3 and 4, we investigate the complexity of Total Dominating Set with no conflicts, Connected Vertex Cover with no conflicts and Steiner Tree with no conflicts for some graph classes. The problems are trivially in NP, hence the NP-completeness proofs will only focus on the NP-hardness of the problems. We aim to prove NP-completeness for the smallest possible classes, for both the support graph and the conflict graph. In the process, we refine results of [DLP16] by drawing a more accurate picture of the NP-completeness of the problems restricted to specific graph classes, both sparse or dense. Since we obtain NP-completeness results for restricted classes of graphs, we introduce a new parameter to capture the locality of conflicts: the *stretch*. In a graph with conflicts (G, C), for any conflict ab of C, the stretch of ab is the distance between a and b in the support graph G. The graph with conflicts (G, C) is of stretch at most k if no conflicts of C are of stretch strictly greater than k. A graph with conflicts (G, C) is of stretch exactly k if any conflict of C is of stretch k.

Most of the following NP-completeness proofs use reductions from a restricted version of 3-SAT. Let us define the problem formally.

- Instance: (X, Cl) where X is a set of boolean variables and Cl a set of disjunctive 3-clauses over X.
- Question: Is there an assignment on X satisfying Cl?

The 3-SAT problem is NP-complete, even if each variable appears in at most 4 clauses [Tov84] (in a positive or a negative form). This result will be useful to reduce the maximum degree of support graph or conflict graph in several reductions. Moreover, one can suppose without loss of generality that each literal appears in at most 3 clauses: otherwise its negation cannot appear and the literal can be set to $true$ and the variable removed from the instance together with clauses in which it appears.

In this paper we need several additional notations and graph classes. If $G = (V, E)$ is a graph and $X \subseteq V$ then $G[X]$ is the *graph induced* by X in G. Two vertices linked by an edge are *neighbors*. The *neighborhood* of u in G is noted $N_G(u)$ or $N(u)$ when there is no ambiguities. The path on n vertices is denoted by P_n.

A graph of n vertices is a *Dirac graph* if each vertex has degree at least $n/2$. A graph is a *split graph* if it can be partitioned into an independent set and a clique. A *caterpillar* is a tree that has a dominating

Support graph G		Conflict graph C		Complexity	Reference
Class	max degree	Class	Stretch		
TOTAL DOMINATING SET (TDSwnC)					
Bipartite	4	Maximum degree 1	$= 2$	NPC	Thm 1
Caterpillar	3	Maximum degree 1	Finite	NPC	Thm 2
-	2	-	-	P	Thm 3
CONNECTED VERTEX COVER (CVCwnC)					
Bipartite	4	Maximum degree 1	$= 2$	NPC	Thm 4
Minimal degree $(1/2 - \epsilon) \cdot n$	-	-	-	NPC	Thm 6
Tree	-	-	Finite	P	Rmk 1
Split	-	-	-	P	Thm 5
STEINER TREE (STwnC)					
Bipartite	4	Maximum degree 1	$= 2$	NPC	Thm 7
Tree	-	-	Finite	P	Rmk 2
Planar bipartite	3	Maximum degree 1	-	NPC	Thm 8
Planar chordal	4	$\bigcup$ complete bipartite ≤ 4	Finite	NPC	Thm 9
Split	-	Maximum degree 1	$= 1$	NPC	Thm 10
Dirac	-	P_1,P_2 and P_3	≤ 2	NPC	Thm 11
Dirac	-	Dirac	≤ 2	NPC	Thm 12
VERTEX COVER (VCwnC)					
-	-	-	-	P	[DLP16]
Finding a path (PwnC)					
-	-	-	-	NPC	[Yin97]
INDEPENDANT DOMINATING SET (IDSwnC)					
Bipartite	5	Maximum degree 1	$= 2$	NPC	[CL]
Path	2	Maximum degree 1	Finite	NPC	[CL]
Path	2	Dirac	Finite	NPC	[CL]
Dirac	-	P_2 and P_4	≤ 2	NPC	[CL]
Dirac	-	Dirac	≤ 2	NPC	[CL]
DOMINATING SET (DwnC)					
Bipartite	4	Maximum degree 1	$= 2$	NPC	[CL]
Path	2	$\bigcup P_3$	Finite	NPC	[CL]
Union of paths	2	Maximum degree 1	-	NPC	[CL]
Path	2	Maximum degree 1	Finite	O(1)	[CL]
Path	2	Dirac	Finite	NPC	[CL]
Dirac	-	Dirac	≤ 2	NPC	[CL]

Fig. 1: Complexity results for problems with conflicts.

path. A *claw* is a star graph with 4 vertices. A graph is *chordal* if every cycle of length at least 4 has a chord. Figure 1 summarizes new results from this paper and from other works to provide a wider picture of the complexity of problems with conflicts.

2 Total Dominating Set

Given (G, C) where $G = (V, E)$ is the support graph and C the conflict graph, a *total dominating set with no conflicts* (TDSwnC) is a subset of vertices $S \subseteq V$ such that:
- for each $x \in V$ $\exists y \in S$ with $xy \in E$
- for each $xy \in C$, $x \notin S$ or $y \notin S$

We first prove a NP-completeness result when the stretch is exactly 2. This result shows that the problem is hard even when the conflicts are very local.

Theorem 1 *Given (G, C) a graph with conflicts, deciding whether there exists a TDSwnC is NP-complete even if G is a bipartite graph of maximum degree 4 and C is a graph of maximum degree 1 and of stretch exactly 2.*

Proof: Let (X, Cl) be a 3-SAT instance where each variable is in at most 4 clauses. Suppose without loss of generality that each literal is in at most 3 clauses. Construct (G, C) an instance of TDSwnC as follows: for each variable $x_i \in X$ vertices $x_i, \bar{x_i}$ and r_i are created along with the edges $x_i r_i$ and $\bar{x_i} r_i$, and the conflict $x_i \bar{x_i}$. For each clause $c_m = (a \vee b \vee c)$ where a, b, c are literals, vertex c_m and edges $c_m a, c_m b, c_m c$ are created. Vertices r_i are of degree 2, vertices c_m of degree 3, and vertices x_i and $\bar{x_i}$ of degree at most 4 (neighbors from r_i and at most 3 clauses). An example is given in Figure 2.

Let A be an assignment on X satisfying Cl. Construct S a TDSwnC of (G, C). For each i, $r_i \in S$. Thus, the only vertices of G not yet dominated are the vertices c_m for each clause, and the vertices r_i. For each $x_i = 1$ of A, set $x_i \in S$. For each $\bar{x_i} = 1$ of A, set $\bar{x_i} \in S$. Since A is an assignment, it does not induce conflicts in S. Moreover, each clause is satisfied, thus each c_m has a neighbor in S. For each variable x_i, either x_i or $\bar{x_i}$ is set to 1, thus each r_i is dominated. Then S is a TDSwnC of (G, C).

Let S be a TDSwnC of (G, C) and let A be the following assignment on X: $x_i = 1$ if $x_i \in S$ and $x_i = 0$ if $\bar{x_i} \in S$. For each vertex c_i there exists a vertex $x_j \in S$ connected to c_i to dominate it, thus the corresponding clause is satisfied by A. Moreover, x_i and $\bar{x_i}$ are in conflict, thus the assignment is consistent. □

We now prove the NP-completeness for caterpillars of maximum degree 3 when the conflict graph is of maximum degree 1. To achieve this result, we first prove a weaker one in Lemma 1 and then in lemmas 2 and 3 we present gadgets to simplify both the support graph and the conflict graph.

Lemma 1 *Given (G, C) a graph with conflicts, deciding whether there exists a TDSwnC is NP-complete even if G is a disjoint union of claws and C is a disjoint union of complete bipartite graphs of at most 4 vertices.*

Proof: Let (X, Cl) be a 3-SAT instance where each variable is in at most 4 clauses. Construct (G, C) an instance of TDSwnC. For each clause $c_i = (a \vee b \vee c)$ where a, b, c are literals, construct a star of center c_i with 3 leaves a, b, c. For each pair $\{a, \bar{a}\}$ of vertices, create the conflict $a\bar{a}$. The graph G is then an union of claws (which are caterpillars of maximum degree 3) and the conflict graph is an union of complete

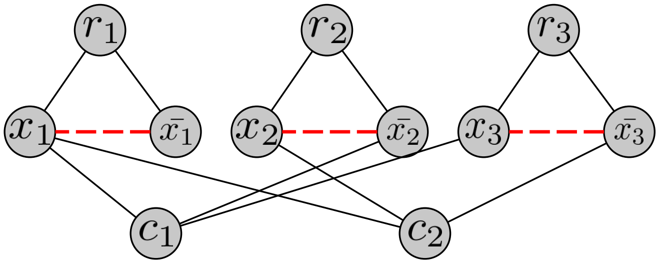

Fig. 2: Reduction from the 3-SAT instance $(x_1 \vee \bar{x_2} \vee x_3) \wedge (x_1 \vee x_2 \vee \bar{x_3})$ to an instance of TDSwnC. The dashed edges denote conflicts.

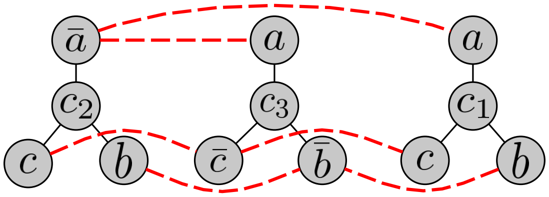

Fig. 3: Graph equivalent to the 3-SAT formula $(\bar{a} \vee b \vee c) \wedge (a \vee \bar{b} \vee \bar{c}) \wedge (a \vee b \vee c)$. The dashed edges denote conflicts.

bipartite graphs of at most 4 vertices (because each variable is in at most 4 clauses). An example of this reduction is shown in Figure 3.

Let A be an assignment on X satisfying Cl. Construct $S = \bigcup_i \{c_i\} \cup P$ where P is the set of vertices corresponding to the positive literals of A. Since A is an assignment, a vertex corresponding to a literal and a vertex corresponding to its negation cannot be in S simultaneously. Thus S is without conflicts. For each claw, leaves are dominated by the center. Moreover, each clause is satisfied thus for each claw, a leaf belongs to S and the center is dominated. Thus S is a TDSwnC of G.

Let S be a TDSwnC of (G, C). Let A be the following assignment : $l_i = 1$ if $l_i \in S$, $l_i = 0$ otherwise. A conflict exists for each pair $a, \bar{a}$, thus a literal and its negation cannot be set simultaneously to 1, hence the assignment is consistent. Moreover, for each clause c_i, there is a claw whose center can only be dominated by a vertex representing one of its literals, thus each clause is satisfied. □

In the above reduction, the conflict graph is a disjoint union of small complete bipartite graphs. The next lemma presents gadgets to decompose these bipartite graphs into graphs of maximum degree 1.

Lemma 2 *Given (G, C) a graph with conflicts where G is a disjoint union of caterpillars of maximum degree $\delta > 1$ and C an union of complete bipartite graph of at most 4 vertices, it is possible to construct (G', C') where G is a disjoint union of caterpillars of maximum degree δ and C a graph of maximum degree 1 such that (G, C) has a TDSwnC if and only if (G', C') has a TDSwnC.*

Proof: Let (G, C) be a graph with conflicts where G is a disjoint union of caterpillars of maximum degree $\delta > 1$ and C a disjoint union of complete bipartite graphs of at most 4 vertices. Split $K_{1,2}$, $K_{1,3}$ and $K_{2,2}$

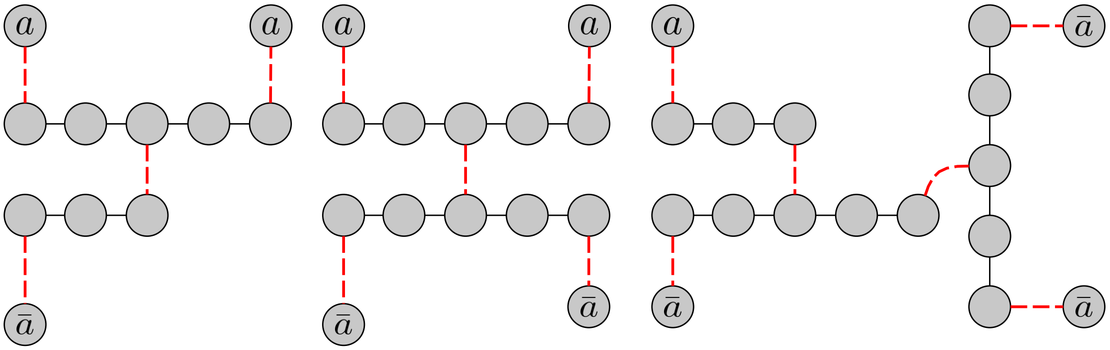

Fig. 4: Gadget used to split $K_{1,2}$(left), $K_{2,2}$ (middle) and $K_{1,3}$ (right) of conflicts into graphs of maximum degree 1. The dashed edges denote conflicts.

of conflicts into graphs of maximum degree 1 using gadgets from Figure 4. One can see by exhaustive search that the choice of a vertex a forbids all the vertices $\bar{a}$ (and conversely), and that the new paths can be dominated regardless of the choice of a or $\bar{a}$. Only paths were created in this transformation, hence the maximum degree of the support graph did not changed and the conflict graph is of maximum degree 1. □

The conflict graph is now of maximum degree 1 but the support graph is not connected. The next result shows how to connect the disjoint caterpillars using gadgets.

Lemma 3 *Given (G,C) a graph with conflicts where G is a disjoint union of caterpillars of maximum degree $\delta > 2$ and C a graph of maximum degree 1, it is possible to construct (G',C') where G' is a connected caterpillar of maximum degree δ and C' a graph of maximum degree 1 such that (G,C) has a TDSwnC if and only if (G',C') has a TDSwnC.*

Proof: Let (G,C) be a graph with conflicts where G is a disjoint union of caterpillars of maximum degree $\delta > 2$ and C is a graph of maximum degree 1. If G is a single caterpillar, the lemma is true. Otherwise, let p_1 and p_2 be two caterpillars of G and let (G_1,C_1) be the graph with conflicts where p_1 and p_2 are connected by p, where p is the caterpillar shown in Figure 5. More specifically, connect vertices 1 and 4 of p to the extremities of the longest path of the caterpillars p_1 and p_2. Thus, the graph is still an union of caterpillars. The extremities of paths are leaves, their degrees change from 1 to 2, hence the maximum degree of the graph does not change. Moreover, p can be dominated only if 1 and 4 do not belong to the TDSwnC. This can be proved by exhaustive search of the TDSwnC of the gadget. Thus it can be used to connect two caterpillars without changing the existence of a solution. Moreover, G_1 has one less caterpillar than G. Repeat this transformation until there is only one caterpillar. □

Theorem 2 *Given (G,C) a graph with conflicts, deciding whether there exists a TDSwnC is NP-complete, even if G is a caterpillar of maximum degree 3 and C a graph of maximum degree 1.*

Proof: Since claws are caterpillars of maximum degree 3, Theorem 2 follows from lemmas 1, 2 and 3. □

Previous results proved that TDSwnC is NP-complete in caterpillars of maximum degree 3. We now prove that this result is in some way the strongest possible, since the problem becomes polynomial when the maximum degree is 2.

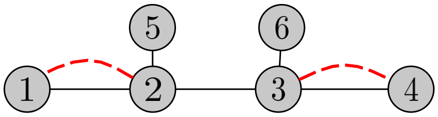

Fig. 5: Gadget used to connect two connected components without changing the existence of a solution. The dashed edges denote conflicts.

Theorem 3 *If G is a graph of maximum degree 2 then deciding whether (G, C) has a TDSwnC can be done in polynomial time.*

Proof: We reduce the problem to 2-SAT for which polynomial-time algorithms are known (see [APT79]). Let $(G = (V, E), C)$ be an instance of TDSwnC where G has maximum degree 2. Suppose without loss of generality that there is no isolated vertex (otherwise the graph has no TDSwnC). Construct (X, Cl) the following SAT instance: $X = V$ and Cl $= \bigwedge_{x\in V}(\bigvee_{y\in N(x)} y) \bigwedge_{ab\in C}(\bar{a} \vee \bar{b})$. Each vertex has at most 2 neighbors hence we obtain a 2-SAT instance.

Let A be an assignment on X satisfying Cl. Then $S = \{x \mid x = 1\}$ is without conflicts in C since for each conflict ab the clause $\bar{a} \vee \bar{b}$ exists. Moreover, for each $x \in V$, the clause $\bigvee_{y\in N(x)} y$ exists, hence x is dominated by one of its neighbors. Thus S is a TDSwnC.

Let S be a TDSwnC of (G, C). Construct A an assignment on X as follows: x is set to 1 if $x \in S$, x is set to 0 otherwise. For each clause c, there exists a vertex of V which can only be dominated by vertices representing literals of the clause. One of these vertices belongs to S, thus the clause is satisfied. For each clause $\bar{a} \vee \bar{b}$, there exists a conflict ab, thus at most one vertex belongs to S, hence the clause is satisfied. Thus A satisfies Cl. □

3 Connected Vertex Cover

Given (G, C) where $G = (V, E)$, a *connected vertex cover with no conflicts* (CVCwnC) is a subset of vertices $S \subseteq V$ such that:
- for each $xy \in E$, $x \in S$ or $y \in S$
- for each $xy \in C$, $x \notin S$ or $y \notin S$
- $G[S]$ is connected.

Theorem 4 *Given (G, C) a graph with conflicts, deciding whether there exists a CVCwnC is NP-complete even if G is a bipartite graph of maximum degree 4 and C is a graph of maximum degree 1 of stretch exactly 2.*

Proof: Let (X, Cl) be a 3-SAT instance where each variable is in at most 4 clauses. Construct (G, C) an instance of CVCwnC as follows: for each variable α, vertices $\alpha, \bar{\alpha}$ and r_α are created, along with the edges αr_α and $\bar{\alpha} r_\alpha$ and the conflict $\alpha\bar{\alpha}$. Vertices r_α are connected by intermediate vertices r_i. For each clause $c_i = (a \vee b \vee c)$, vertices c_i and c'_i are created, along with the edges $c_i c'_i, c_i a, c_i b, c_i c$. Thus G is bipartite of maximum degree 4 and conflicts are a graph of maximum degree 1 of stretch exactly 2. An example is shown in Figure 6.

Let A be an assignment on X satisfying Cl. Let S be the set of vertices corresponding to the positive literals of A. Vertices r_α and r_i and c_i are also added to S. Thus S is a vertex cover. Moreover, it is

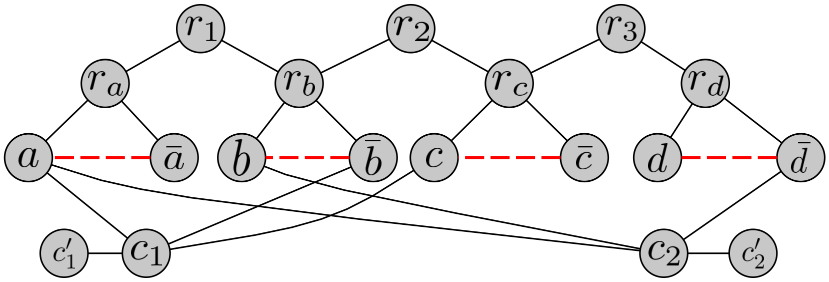

Fig. 6: Graph equivalent to the 3-SAT formula $(a \vee \bar{b} \vee c) \wedge (a \vee b \vee \bar{d})$. The dashed edges denote conflicts.

without conflicts since a literal and its negation cannot be set to 1 simultaneously. Vertices r_α, r_i and vertices corresponding to literals are connected, and for each clause $c_i = (a \vee b \vee c)$, a, b or c belong to S, thus c_i is connected.

Let S be a CVCwnC of (G, C). Literals corresponding to vertices of S are set to 1, other variables are set to 0. Since conflicts exist between each literal and its negation, we obtain an assignment. Moreover, vertices c_i necessarily belong to S, and each needs to be connected by a vertex corresponding to a literal of the associated clause. Thus, each clause is satisfied. □

Let us point out two graph classes in which deciding the existence of a CVCwnC can be done in polynomial time.

Remark 1 *Deciding the existence of CVCwnC is polynomial in trees. There exists a unique connected vertex cover minimal for inclusion, the set of internal vertices of the tree: it is sufficient to test if it is without conflicts.*

Theorem 5 *Given (G, C) a graph with conflicts, deciding whether there exists a CVCwnC can be done in polynomial time if G is a split graph.*

Proof: Let $G = (V, E)$ be a split graph where $V = K \cup I$ where K is a clique and I is an independent set and C the conflict graph.

- If K is without conflicts, then K is a CVCwnC of (G, C).

- If $C[K]$ is a star of center a, then if $S = (K - a) \cup N_G(a)$ is without conflicts, S is a CVCwnC. (Two possibilities for the center a if the star is an edge). If $(K - a) \cup N_G(a)$ has a conflict, then (G, C) does not have a CVCwnC.

- If $C[K]$ is not a star, then at least 2 vertices a and b of K cannot be in the solution: the edge ab will not be covered and thus there is no CVCwnC. □

In Theorem 4 we proved NP-completeness for bipartite graphs. We are now interested in dense graphs, and prove NP-completeness for dense graphs.

Theorem 6 *For all $\epsilon > 0$, given (G, C) a graph with conflicts, deciding whether there exists a CVCwnC is NP-complete even if G is of minimum degree $(1/2 - \epsilon)n$.*

Proof: Let (X, Cl) be a 3-SAT instance of m clauses over n variables. Construct (G, C) an instance of CVCwnC as follows. Let d be the least integer greater than $(m + 2n)/2\epsilon$. For each variable $x \in X$ create

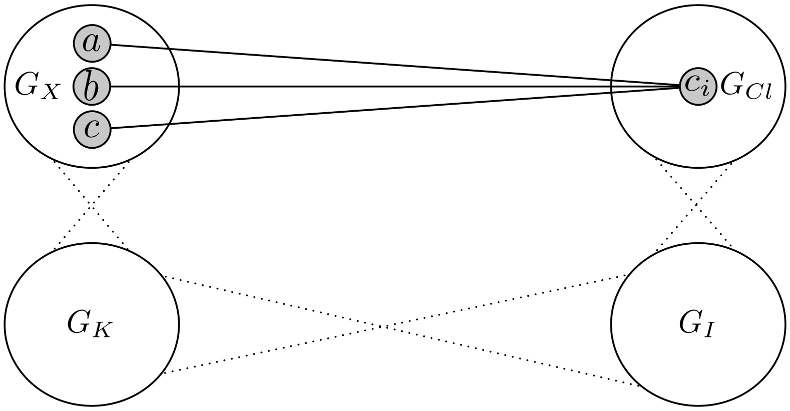

Fig. 7: Construction of the dense graph. Dotted lines represent complete bipartites between subsets of vertices.

the vertices $x, \bar{x}$. For each clause c_i create vertex c_i. Let G_X be the set of vertices representing literals. The graph G_X has $2n$ vertices. Let G_{Cl} be the set of m vertices representing clauses. Construct G_K a clique of size d and G_I an independent set of size d. Add edges such that each vertex of G_K is connected to each vetex of G_X, each vertex of G_{Cl} is connected to each vertex of G_I and each vertex of G_K is connected to each vertex of G_I. For each clause $c_i = (a \vee b \vee d)$ edges $c_i a, c_i b, c_i d$ are created. Construct C as follow. Each pair $\{x, \bar{x}\}$ is in conflict. Each vertex of G_I is in conflict with all the other vertices of G. We present a scheme of the support graph used in the reduction in Figure 7. To ensure readability, conflicts are not represented.

The graph G has $m + 2n + 2d$ vertices and $d \geq (m + 2n)/2\epsilon$. Because of the complete bipartite graphs, vertices of G_X, G_{Cl}, G_I and G_K have degree at least d. It can be derived by basic arithmetic on inequalities that $d > (1/2 - \epsilon)(m + 2n + 2d)$, hence the graph satisfies the assumption of the theorem.

Let A be an assignment on X satisfying Cl, and let A_1 be the set of positive literals of A. Set VC $=$ $A_1 \cup G_K \cup G_{\text{Cl}}$. The set VC does not contain simultaneously vertices representing a literal and its negation, neither vertices of G_I, thus it is a set without conflicts in C. Edges between G_K and G_X, G_{Cl} and G_I, G_K and G_I, G_X and G_{Cl} are covered. Thus VC is a vertex cover of G. Vertices of G_K and G_X are connected. Moreover, for each vertex $c_i \in$ Cl there exists a vertex x of A_1 corresponding to a positive literal of c_i. Thus VC is connected.

Let VC be a CVCwnC of (G, C). Let A be the following assignment: for each literal x, if x is represented by a vertex of VC, then $x = 1$, otherwise, $x = 0$. The set VC is not reduced to a single vertex, hence VC cannot contain vertices of G_I because of conflicts between G_I and all the vertices. The set VC must contain the neighborhood of G_I: $G_{\text{Cl}} \cup G_K$. Moreover, each vertex of G_{Cl} must be connected to G_K. It can only be done via vertices of G_X. For each vertex representing a clause $(a \vee b \vee c)$ there exists a vertex representing one of its literals in VC. Hence, A satisfies Cl. Moreover, a vertex representing a variable and a vertex representing its negation cannot be set to 1 simultaneously, thus the assignment is consistent. □

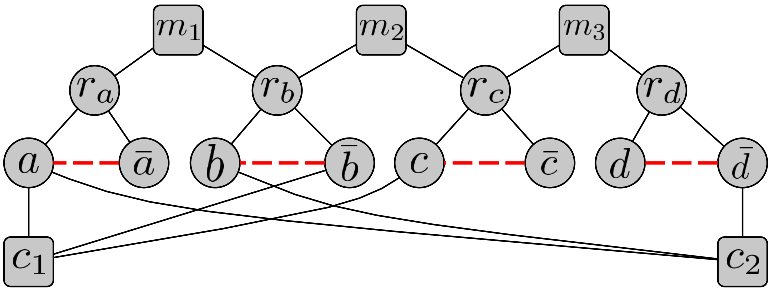

Fig. 8: Graph equivalent to the 3-SAT formula $(a \vee \bar{b} \vee c) \wedge (a \vee b \vee \bar{d})$. Vertices of M are squares. The dashed edges denote conflicts.

4 Steiner tree

A Steiner tree of $(G = (V, E), M)$ where $M \subseteq V$ is a subtree of G that includes all the vertices of M.

Given (G, M, C) where $G = (V, E)$ and $M \subseteq V$, a *Steiner Tree without conflicts* (STwnC) is a subset of vertices $S \subseteq V$ such that :

- for each $xy \in C$, $x \notin S$ or $y \notin S$
- $M \subseteq S$
- $G[S]$ is connected.

If $G[S]$ is connected and without conflicts, it is easy to extract a covering tree of S, hence we are interested only in the set of vertices of the tree.

First, we prove NP-completeness when the conflicts are local.

Theorem 7 *Given (G, M, C) a graph with conflicts and a subset of vertices, deciding whether there exists a STwnC is NP-complete even if G is a bipartite graph of maximum degree 4 and C a graph of maximum degree 1 and of stretch exactly 2.*

Proof: Let (X, Cl) be a 3-SAT instance where each variable is in at most 4 clauses. Suppose without loss of generality that each literal is in at most 3 clauses. Construct (G, M, C) an instance of STwnC as follows: for each variable α vertices $\alpha, \bar{\alpha}$ and r_α are created, along with the edges αr_α and $r_\alpha \bar{\alpha}$ and the conflict $\alpha\bar{\alpha}$. Vertices r_α are connected by intermediate vertices m_i. For each clause $c_i = (a \vee b \vee c)$, a vertex c_i is created along with the edges $c_i a, c_i b, c_i c$. The graph is of maximum degree 4, and the conflict graph is a graph of maximum degree 1 and of stretch exactly 2. The set M is composed of all the vertices m_i and c_i. An example is shown in Figure 8.

Let A be an assignment on X satisfying Cl. Set $S = M \bigcup_\alpha \{r_\alpha\} \cup P$ where P is the set of vertices representing positive literals of A. Since a variable cannot be set to 1 simultaneously with its negation, S is without conflicts. Vertices m_i and r_α are connected. Moreover, for each vertex c_i, there exists a clause c_i in which one of the literals is set to 1 in A, thus c_i is connected to the other vertices of S. Thus, S is connected in G.

Let S be a STwnC of (G, M, C). The value of literals of X which vertices are in S is set to 1. Since conflicts exist between variables and their negation, we obtain an assignment. Moreover, for each vertex c_i representing a clause $a \vee b \vee c$, one of the vertices a, b, c must belong to S to ensure connectivity with m_j vertices of M, hence the clause is satisfied. □

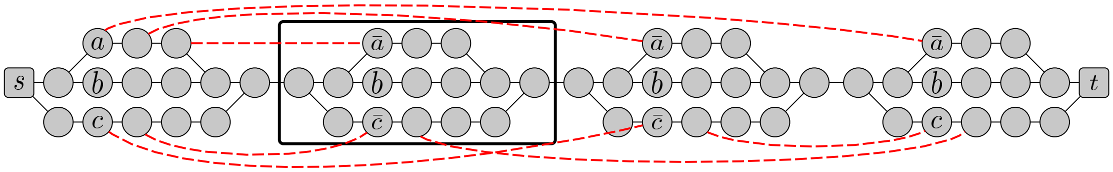

Fig. 9: Graph equivalent to the 3-SAT formula $(a \vee b \vee c) \wedge (\bar{a} \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee \bar{c}) \wedge (\bar{a} \vee b \vee c)$. Vertices of M are squares. In the black rectangle, gadget corresponding to the clause $(\bar{a} \vee b \vee \bar{c})$. The dashed edges denote conflicts.

The next theorems are NP-completeness results for more restricted class of support graph, but the conflicts are no longer local (i.e., stretch is not bounded).

Theorem 8 *Given (G, M, C) a graph with conflicts and a subset of vertices, deciding whether there exists a STwnC is NP-complete even if G is a planar bipartite graph of maximum degree 3 and C a graph of maximum degree 1.*

Proof: Let (X, Cl) be a 3-SAT instance where each variable is in at most 4 clauses. Construct (G, M, C) an instance of STwnC where G is a planar bipartite graph of maximum degree 3 and C a graph of maximum degree 1 such that there exists an assignment on X satisfying Cl if and only if there exists a STwnC of (G, M, C).

For each clause, create a gadget of 15 vertices composed of 3 non-disjoint paths having the same extremities, each corresponding to a literal. One can see this gadget in Figure 9. These gadgets are connected linearly, and the set M, composed of only two vertices, is the first vertex of the first gadget and the last vertex of the last gadget. For each pair of literals $\{x, \bar{x}\}$, add a conflict between a vertex with no conflicts of the path representing x and a vertex with no conflicts of the path representing $\bar{x}$. The graph is planar bipartite of maximum degree 3 and the conflict graph is of maximum degree 1.

Let A be an assignment on X satisfying Cl. Construct S a subset of G as follows. For each gadget representing a clause $(a \vee b \vee c)$, choose in S a path representing a positive literal. Each gadget is passed through, hence vertices of M are connected. The set S is the path of vertices connecting the two vertices of M. Moreover, this path does not pass through paths representing variable and their negation, hence it is without conflicts.

Let S be a STwnC of (G, M, C). Suppose without loss of generality that S is minimal for inclusion (otherwise, it can be minimalized). Construct A an assignment on X satisfying Cl as follows: for each gadget representing a clause $(a \vee b \vee c)$, S contains a path representing one of its literals. This one is set to 1 in A. Since there exist conflicts between paths representing variable and their negation, the assignment A is consistent. Moreover, for each clause, a literal is set to 1 hence A satisfies Cl. □

Theorem 9 *Given (G, M, C) a graph with conflicts and a subset of vertices, deciding whether there exists a STwnC is NP-complete even if G is a planar chordal graph of maximum degree 4 and C a disjoint union of complete bipartite graphs of at most 4 vertices.*

Proof: Let (X, Cl) be a 3-SAT instance where each variable is in at most 4 clauses. Construct (G, M, C) the following instance of STwnC. For each clause $c_i = (a \vee b \vee c)$, vertices c_i, c'_i, a, b, c, m_i and edges $c'_i c_i$,$c_i a$,$c_i b$, $c_i c$, ab, ac, am_i, bm_i, bc, cm_i are created. The set M is the set of all vertices m_i and c'_i. Vertices c'_i are connected to form a path. For each pair of vertices $\{\alpha, \bar{\alpha}\}$ representing a literal and its

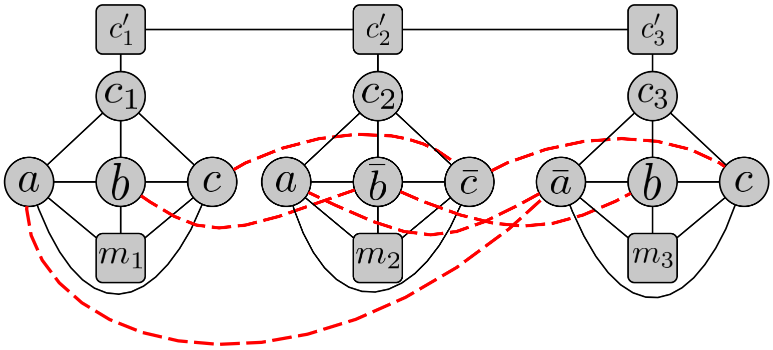

Fig. 10: Graph equivalent to the 3-SAT formula $(a \vee b \vee c) \wedge (a \vee \bar{b} \vee \bar{c}) \wedge (\bar{a} \vee b \vee c)$. Vertices of M are squares. The dashed edges denote conflicts.

negation, the conflict $\alpha\bar{\alpha}$ is created. The graph is chordal planar of maximum degree 4, and the conflict graph is an union of complete bipartite graphs of at most 4 vertices. An example is shown in Figure 10.

Let A be an assignment on X satisfying Cl. Construct $S = M \bigcup_i \{c_i\} \cup P$ where P is the set of vertices representing positive literals of A. Then $\bigcup_i \{c_i\} \bigcup_i \{c'_i\}$ is connected. Moreover, for each clause c_i, a vertex x corresponding to a positive literal belongs to S. Hence, the vertex m_i is connected and S is connected in G. Moreover, conflicts are only between variables and their negation, which can not be simultaneously set to 1 in A. Thus, S is without conflicts.

Let S be a STwnC of (G, M, C). Literals corresponding to vertices of S are set to 1, the other variables are set to 0. Since there exist conflicts between vertices representing variables and their negation, we obtain an assignment. Moreover, each vertex m_i must be connected to the other vertices of S by a vertex representing a literal of the associated clause, which is satisfied. □

Remark 2 *Deciding the existence of a STwnC is polynomial in trees. There exists an unique Steiner tree minimal for inclusion, hence it is sufficient to test if it is without conflicts.*

Theorem 10 *Given (G, M, C) a graph with conflicts and a subset of vertices, deciding whether there exists a STwnC is NP-complete even if G is a split graph and C a graph of maximum degree 1 and of stretch 1.*

Proof: Let (X, Cl) be a 3-SAT instance. Suppose without loss of generality that Cl contain several clauses. Create $(G = (K, I, E), M, C)$ an instance of STwnC as follows. Each literal becomes a vertex of K and each clause a vertex of I. Conflicts are added between variable and their negations, and edges between clauses and their literals. The graph is a split graph, and the conflict graph is a graph of maximum degree 1 and of stretch 1. Set $M = I$. An example is shown in Figure 11.

Let A be an assignment on X satisfying Cl. Construct $S = I \cup P$ where P is the set of vertices corresponding to positive literals of A. Since a literal and its negation cannot both be set to 1, S is without conflicts. Moreover, for each $c_i \in I$, the associated clause is satisfied, thus there exists a neighbor in $K \cap S$. Hence, each vertex of I is connected to $K \cap S$ and since K is a clique, S is connected.

Let S be a STwnC of $(G = (K, I, E), M, C)$. The value of literals corresponding to vertices of S is set to 1, and the value 0 is set to other variables. Since there exist conflicts between vertices representing

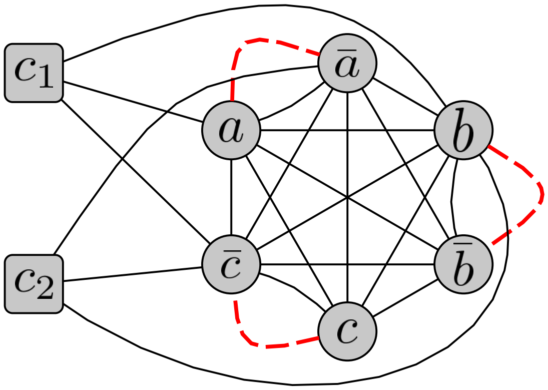

Fig. 11: Graph equivalent to the 3-SAT formula $(a \vee \bar{c} \vee b) \wedge (\bar{a} \vee b \vee \bar{c})$. Vertices of M are squares. The dashed edges denote conflicts.

literals and vertices representing their negation, we have an assignment. Moreover, each vertex c_i must be connected to K by a vertex representing a literal of the associated clause, thus the clause is satisfied. □

Previous theorems proved NP-completeness for sparse (or split) support and conflict graphs. We now prove that the problem remains NP-complete in a class of dense graphs: Dirac graphs (which are graphs of minimum degree $n/2$).

Theorem 11 *Given (G, M, C) a graph with conflicts and a subset of vertices, deciding whether there exists a STwnC is NP-complete even if G is a Dirac graph and C a disjoint union of P_1, P_2 and P_3.*

Proof: Let $(G_1 = (V_1, E_1), M_1, C_1)$ be an instance of STwnC where C_1 is a graph of maximum degree 1. This problem is NP-complete from Theorem 7. Construct $(G_2 = (V_2, E_2), M_2, C_2)$ a graph with conflicts such that there exists a STwnC of (G_2, M_2, C_2) if and only if there exists a STwnC of (G_1, M_1, C_1). Let $n = |V_1|$. Set $V_2 = V_1 \cup A \cup B$ where A is a path of length n and B an independent set of size $2n$. $M_2 = M_1 \cup A$. A is connected to an arbitrary vertex of M_1. A complete bipartite graph is added between B and $V_2 - B$. Conflicts of C_1 are added to C_2. Moreover, each vertex of A is in conflict with 2 distinct vertices of B. By construction, G_2 is a Dirac graph and C_2 a disjoint union of P_1, P_2 and P_3.

Let T_1 be a STwnC of G_1. Then $T_2 = T_1 \cup A$ is a STwnC of G_2.

Suppose there exists T_2 a STwnC of (G_2, M_2, C_2). Then $T_2 \cap B = \emptyset$. Moreover, vertices of A do not connect vertices of V_1. Hence, $T_1 = T_2 - A$ is a STwnC of (G_1, M_1, C_1). □

Theorem 12 *Given (G, M, C) a graph with conflicts and a subset of vertices, deciding whether there exists a STwnC is NP-complete even if G is a Dirac and C a Dirac graph.*

Proof: Let $(G_1 = (V_1, E_1), M_1, C_1)$ be an instance of STwnC. Construct $(G_2 = (V_2, E_2), M_2, C_2)$ a Dirac graph with conflicts such that there exists a STwnC of (G_2, M_2, C_2) if and only if there exists a STwnC of (G_1, M_1, C_1). Let $n = |V_1|$. Set $V_2 = V_1 \cup A$ where A is an independent set of size n. $M_2 = M_1$. A complete bipartite graph is created between V_1 and A. Conflicts of C_1 exist in C_2.

Moreover, each vertex of A is in conflict with all vertices of V_1. By construction, G_2 and C_2 are Dirac graphs.

Let T be a STwnC of (G_1, M_1, C_1). Thus it is a STwnC of G_2.

Suppose that there exists T a STwnC of (G_2, M_2, C_2). Then $T \cap A = \emptyset$. Hence, T is a STwnC of (G_1, M_1, C_1). □

5 Conclusion

Our paper strengthens the results of [DLP16] for CVCwnC and STwnC, by proving NP-completeness of deciding their existence in smaller graph classes. More exactly, we proved that these problems remain NP-complete for some classes of sparse graphs. For STwnC, we proved that the problem is also NP-complete in dense graphs. We also extended the notion of conflicts to a new problem. We proved NP-completeness of deciding the existence of a TDSwnC in a very sparse graph class: caterpillars of maximum degree 3. Moreover, we proved that this result is in some way the strongest possible since the problem becomes polynomial in graph of maximum degree 2. Furthermore, we proved that all these problems are NP-complete even when the stretch of the conflicts is two at most, i.e., when the conflicts are local.

A natural extension of this work would be to work on the complexity of CVCwnC and TDSwnC in dense graphs, and the complexity of CVCwnC in other classes of sparse graphs, for example planar graphs. It would also be interesting to study other graph problems with conflicts.

Acknowledgements

We would like to thank the anonymous reviewers for their usefull remarks which helped us to improve the writing of this paper.

References

[ACG+12] Giorgio Ausiello, Pierluigi Crescenzi, Giorgio Gambosi, Viggo Kann, Alberto Marchetti-Spaccamela, and Marco Protasi. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 2012.

[APT79] Bengt Aspvall, Michael F Plass, and Robert Endre Tarjan. A linear-time algorithm for testing the truth of certain quantified boolean formulas. *Information Processing Letters*, 8(3):121–123, 1979.

[CL] Alexis Cornet and Christian Laforest. Domination problems with conflicts. *Submited*.

[DLP16] Francois Delbot, Christian Laforest, and Raksmey Phan. Hardness Results and Approximation Algorithms for Discrete Optimization Problems with Conditional and Unconditional Forbidden Vertices. Research report, HAL, https://hal.archives-ouvertes.fr/hal-01257820, January 2016.

[GMO76] Harold N Gabow, Shachindra N Maheshwari, and Leon J Osterweil. On two problems in the generation of program test paths. *IEEE Transactions on Software Engineering*, (3):227–231, 1976.

[KLM13a] Mamadou Moustapha Kanté, Christian Laforest, and Benjamin Momège. An exact algorithm to check the existence of (elementary) paths and a generalisation of the cut problem in graphs with forbidden transitions. In *SOFSEM 2013*, volume LNCS 7741, pages 257–267, 2013.

[KLM13b] Mamadou Moustapha Kanté, Christian Laforest, and Benjamin Momège. Trees in graphs with conflict edges or forbidden transitions. In *TAMC 2013, Hong Kong*, volume LNCS 7876, pages 343–354, 2013.

[KMMN15] Mamadou Moustapha Kanté, Fatima Zahra Moataz, Benjamin Momège, and Nicolas Nisse. Finding paths in grids with forbidden transitions. In *Graph-Theoretic Concepts in Computer Science - 41st International Workshop, WG 2015*, volume LNCS 9224, pages 154–168, 2015.

[Kov13] Jakub Kováč. Complexity of the path avoiding forbidden pairs problem revisited. *Discrete Applied Mathematics*, 161(10):1506–1512, 2013.

[KP09] Petr Kolman and Ondřej Pangrác. On the complexity of paths avoiding forbidden pairs. *Discrete Applied Mathematics*, 157(13):2871–2876, 2009.

[LM14] Christian Laforest and Benjamin Momège. Some hamiltonian properties of one-conflict graphs. In *IWOCA 2014*, volume LNCS 8986, pages 262–273, 2014.

[LM15] Christian Laforest and Benjamin Momège. Nash-Williams-type and Chvátal-type conditions in one-conflict graphs. In *SOFSEM 2015*, volume LNCS 8939, pages 327–338, 2015.

[Tov84] Craig A. Tovey. A simplified NP-complete satisfiability problem. *Discrete Applied Mathematics*, 8(1):85 – 89, 1984.

[Yin97] Hananya Yinnone. On paths avoiding forbidden pairs of vertices in a graph. *Discrete Applied Mathematics*, 74(1):85–92, 1997.