Complexity of Unique (Optimal) Solutions in Graphs: Vertex Cover and Domination

Olivier Hudry
LTCI, Télécom ParisTech, Université Paris-Saclay
46 rue Barrault, 75634 Paris Cedex 13 - France
& Antoine Lobstein
Centre National de la Recherche Scientifique
Laboratoire de Recherche en Informatique, UMR 8623,
Université Paris-Sud, Université Paris-Saclay
Bâtiment 650 Ada Lovelace, 91405 Orsay Cedex - France

olivier.hudry@telecom-paristech.fr, antoine.lobstein@lri.fr

Abstract

We study the complexity of four decision problems dealing with the *uniqueness* of a solution in a graph: "Uniqueness of a Vertex Cover with bounded size" (U-VC) and "Uniqueness of an Optimal Vertex Cover" (U-OVC), and for any fixed integer $r \geq 1$, "Uniqueness of an r-Dominating Code with bounded size" (U-DC$_r$) and "Uniqueness of an Optimal r-Dominating Code" (U-ODC$_r$). In particular, we give a polynomial reduction from "Unique Satisfiability of a Boolean formula" (U-SAT) to U-OVC, and from U-SAT to U-ODC$_r$. We prove that U-VC and U-DC$_r$ have complexity equivalent to that of U-SAT (up to polynomials); consequently, these problems are all *NP*-hard, and U-VC and U-DC$_r$ belong to the class *DP*.

Key Words: Graph Theory, Complexity Theory, *NP*-Hardness, Decision Problems, Polynomial Reduction, Uniqueness of (Optimal) Solution, Domination, Dominating Codes, Vertex Covers, Boolean Satisfiability Problems

JCMCC **110** (2019), pp. 217-240

1 Introduction

1.1 The Vertex Cover and Domination Problems

For the vast topic of 1-domination in graphs, see [10].

We shall denote by $G = (V, E)$ a finite, simple, undirected graph with vertex set V and edge set E, where an *edge* between $x \in V$ and $y \in V$ is indifferently denoted by xy or yx. The *order* of the graph is its number of vertices, $|V|$. In a connected graph G, we can define the *distance* between any two vertices x and y, denoted by $d_G(x, y)$, as the length of any shortest path between x and y. This definition can be extended to disconnected graphs, using the convention that $d_G(x, y) = +\infty$ if no path exists between x and y. The subscript G can be dropped when there is no ambiguity.

Given a graph $G = (V, E)$, an *independent set*, or *stable set*, is a subset $V^* \subseteq V$ such that for all $u \in V^*$, $v \in V^*$, we have $uv \notin E$. A *clique*, or *complete graph*, is any subgraph $(V^- \subseteq V, E^- \subseteq E)$ such that for all $u \in V^-$, $v \in V^-$, $u \neq v$, we have $uv \in E^-$. For an integer $r \geq 1$, the *r-th power* of G is the graph $G^r = (V, E^r)$, with $E^r = \{uv : u \in V, v \in V, d_G(u, v) \leq r\}$.

A *vertex cover* of G (VC for short) is a subset of vertices $V^* \subseteq V$ such that for every edge $e = uv \in E$, $V^* \cap \{u, v\} \neq \emptyset$. We denote by $\phi(G)$ the smallest cardinality of a VC of G, and call it the *vertex cover number* of G; any VC V^* with $|V^*| = \phi(G)$ is said to be *optimal*.

For any vertex $v \in V$, the *open neighbourhood* $N(v)$ *of* v consists of the set of vertices adjacent to v, i.e., $N(v) = \{u \in V : uv \in E\}$; the *closed neighbourhood of* v is $B_1(v) = N[v] = N(v) \cup \{v\}$. This notation can be generalized to any integer $r \geq 0$ by setting

$$B_r(v) = \{x \in V : d(x, v) \leq r\}.$$

For $X \subseteq V$, we denote by $B_r(X)$ the set of vertices within distance r from X:

$$B_r(X) = \cup_{x \in X} B_r(x).$$

Whenever two vertices x and y are such that $x \in B_r(y)$ (which is equivalent to $y \in B_r(x)$), we say that x and y *r-dominate* each other; note that every vertex r-dominates itself. A set W is said to r-dominate a set Z if every vertex in Z is r-dominated by at least one vertex of W, or equivalently: $Z \subseteq B_r(W)$.

A *code* C is simply a subset of V, and its elements are called *codewords*.

We say that C is an *r-dominating code* in G if all the sets $B_r(v) \cap C$, $v \in V$, are nonempty; in other words, every vertex is r-dominated by C, or $V = B_r(C)$. We denote by $\gamma_r(G)$ the smallest cardinality of an r-dominating code in G, and call it the *r-domination number* of G; any

r-dominating code C with $|C| = \gamma_r(G)$ is said to be *optimal.* The following result needs no proof.

Lemma 1 *For all $r \geq 1$, the code C is r-dominating in G if and only if it is 1-dominating in the r-th power of G.* ◊

The following two problems are well known in graph theory as well as in complexity theory, specially when $r = 1$ for the second problem, stated here for any fixed integer $r \geq 1$.

Problem VC (Vertex Cover with bounded size):
Instance: A graph G and an integer k.
Question: Does G admit a vertex cover of size at most k?

Problem DC_r (r-Dominating Code with bounded size):
Instance: A graph G and an integer k.
Question: Does G admit an r-dominating code of size at most k?

As we shall see, these problems are *NP*-complete (Propositions 7 from [19], [8] and 17 from [8], [14]). In this paper, we wish to locate, in the hierarchy of complexity classes, the following four problems, dealing with the *uniqueness* of solutions, optimal or not.

Problem U-VC (Unique Vertex Cover with bounded size):
Instance: A graph G and an integer k.
Question: Does G admit a *unique* vertex cover of size at most k?

Problem U-OVC (Unique Optimal Vertex Cover):
Instance: A graph G.
Question: Does G admit a *unique optimal* vertex cover?

Problem U-DC_r (Unique r-Dominating Code with bounded size):
Instance: A graph G and an integer k.
Question: Does G admit a *unique* r-dominating code of size at most k?

Problem U-ODC_r (Unique Optimal r-Dominating Code):
Instance: A graph G.
Question: Does G admit a *unique optimal* r-dominating code?

In Sections 2 and 3, we establish our results on vertex covers and dominating codes, respectively; we prove in particular that there is a polynomial reduction from "Unique Satisfiability of a Boolean formula" (U-SAT) to U-OVC, and from U-SAT to U-ODC_r; and that U-VC and U-DC_r are equivalent to U-SAT (up to polynomials). This implies that:

- U-VC, U-OVC, UDC_r and U-ODC_r ($r \geq 1$) are *NP*-hard;
- U-VC and UDC_r ($r \geq 1$) belong to the class *DP*.

We also show that U-OVC and U-ODC_r ($r \geq 1$) belong to the class L^{NP}.

In forthcoming papers, we likewise investigate the issue of the uniqueness of solutions for (a) Boolean satisfiability and graph colouring [15], of which we shall use some of the results in the present paper; (b) Hamiltonian Cycle [16]; (c) r-Identifying Code and r-Locating-Dominating Code [17].

In [14], we already investigated the complexity of the existence of, and of the search for, optimal r-dominating codes, as well as optimal r-dominating codes containing a given subset of vertices; some results will be re-used here, see, e.g., Lemma 16.

For other works in this area, see [11] for vertex covers, and [7], [13], [20] and [21] for some problems related to domination in the binary hypercube.

In the sequel, we shall also need the following tools, which constitute classical definitions and decision problems, related to Boolean satisfiability. We consider a set $\mathcal{X}$ of n *Boolean variables* x_i and a set $\mathcal{C}$ of m *clauses* ($\mathcal{C}$ is also called a *Boolean formula*); each clause c_j contains κ_j *literals*, a literal being a variable x_i or its complement $\overline{x}_i$. A *truth assignment* for $\mathcal{X}$ sets the variable x_i to TRUE, also denoted by T, and its complement to FALSE (or F), or *vice-versa*. A truth assignment is said to *satisfy* the clause c_j if c_j contains at least one true literal, and to satisfy the set of clauses $\mathcal{C}$ if every clause contains at least one true literal. The following decision problems, for which the size of the instance is polynomially linked to $n+m$, are classical problems in complexity.

Problem SAT (Satisfiability):
Instance: A set $\mathcal{X}$ of variables, a collection $\mathcal{C}$ of clauses over $\mathcal{X}$, each clause containing at least two different literals.
Question: Is there a truth assignment for $\mathcal{X}$ that satisfies $\mathcal{C}$?

The following problem is stated for any fixed integer $k \geq 2$.

Problem k-SAT (k-Satisfiability):
Instance: A set $\mathcal{X}$ of variables, a collection $\mathcal{C}$ of clauses over $\mathcal{X}$, each clause containing exactly k different literals.
Question: Is there a truth assignment for $\mathcal{X}$ that satisfies $\mathcal{C}$?

Problem 1-3-SAT (One-in-Three Satisfiability):
Instance: A set $\mathcal{X}$ of variables, a collection $\mathcal{C}$ of clauses over $\mathcal{X}$, each clause containing exactly three different literals.
Question: Is there a truth assignment for $\mathcal{X}$ such that each clause of $\mathcal{C}$ contains *exactly one* true literal?

We shall say that a clause (respectively, a set of clauses) is 1-3-*satisfied* by an assignment if this clause (respectively, every clause in the set) contains exactly one true literal. We shall also consider the following variants of the above problems:

U-SAT (Unique Satisfiability),

U-k-SAT (Unique k-Satisfiability),
U-1-3-SAT (Unique One-in-Three Satisfiability).
They have the same instances as SAT, k-SAT and 1-3-SAT respectively, but now the question is "Is there a *unique* truth assignment...?".

We shall give in Proposition 3 and Corollary 4 what we need to know about the complexities of these problems. We now provide the necessary definitions and notation for complexity.

1.2 Necessary Notions in Complexity

We expound here, not too formally, the notions of complexity that will be needed in the sequel. We refer the reader to, e.g., [1], [8], [18] or [22] for more on this topic.

A *decision problem* is of the type "Given an instance I and a property $\mathcal{PR}$ on I, is $\mathcal{PR}$ true for I?", and has only two solutions, "yes" or "no". The class P will denote the set of problems which can be solved by a *polynomial* (time) algorithm, and the class NP the set of problems which can be solved by a *nondeterministic polynomial* algorithm. A *polynomial reduction* from a decision problem π_1 to a decision problem π_2 is a polynomial transformation that maps any instance of π_1 into an "equivalent" instance of π_2, that is, an instance of π_2 admitting the same answer as the instance of π_1; in this case, we shall write $\pi_1 \rightarrow_p \pi_2$. Cook [4] proved that there is one problem in NP, namely SAT, to which every other problem in NP can be polynomially reduced. Thus, in a sense, SAT is the "hardest" problem inside NP. Other problems share this property in NP and are called *NP-complete* problems; their class is denoted by *NP-C*. The way to show that a decision problem π is NP-complete is, once it is proved to be in NP, to choose some NP-complete problem π_1 and to polynomially reduce it to π. From a practical viewpoint, the NP-completeness of a problem π implies that we do not know any polynomial algorithm solving π, and that, under the assumption $P \neq NP$, which is widely believed to be true, no such algorithm exists: the time required can grow exponentially with the size of the instance (when the instance is a graph, its size is polynomially linked to the order of the graph).

The *complement* of a decision problem, "Given I and $\mathcal{PR}$, is $\mathcal{PR}$ true for I?", is "Given I and $\mathcal{PR}$, is $\mathcal{PR}$ false for I?". The class *co-NP* (respectively, *co-NP-C*) is the class of the problems which are the complement of a problem in NP (respectively, *NP-C*).

For problems which are not necessarily decision problems, a *Turing reduction* from a problem π_1 to a problem π_2 is an algorithm $\mathcal{A}$ that solves π_1 using a (hypothetical) subprogram $\mathcal{S}$ solving π_2 such that, if $\mathcal{S}$ were a polynomial algorithm for π_2, then $\mathcal{A}$ would be a polynomial algorithm for π_1. Thus, in this sense, π_2 is "at least as hard" as π_1. A problem π is *NP-*

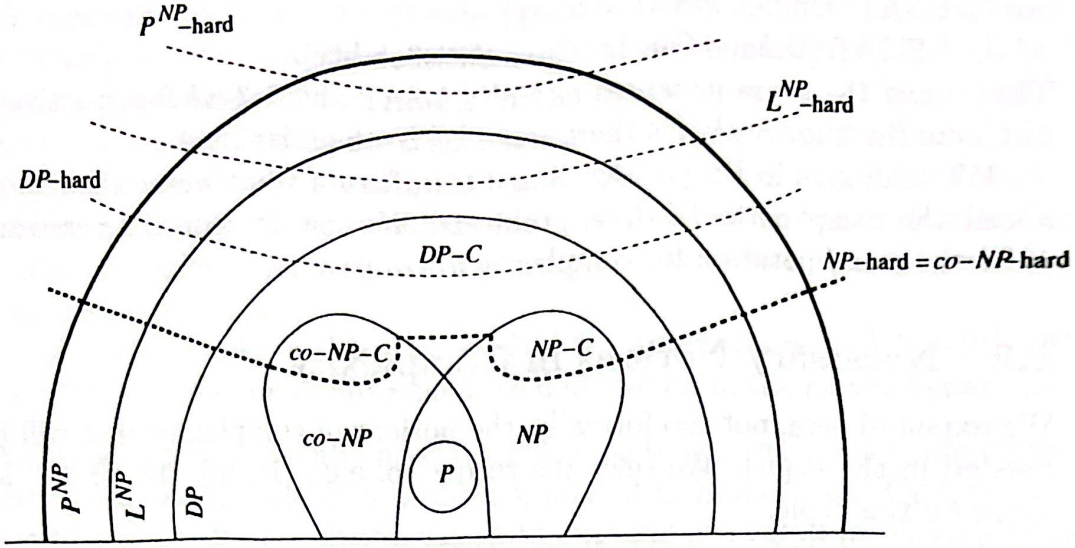

Figure 1: Some classes of complexity.

hard (respectively, *co-NP-hard*) if there is a Turing reduction from some *NP*-complete (respectively, co-*NP*-complete) problem to π [8, p. 113].

Remark 2 *Note that with these definitions, NP-hard and co-NP-hard coincide [8, p. 114].*

The notions of completeness and hardness can of course be extended to classes other than *NP* or co-*NP*. *NP*-hardness is defined differently in [5] and [12]: there, a problem π is *NP-hard* if there is a *polynomial* reduction from some *NP*-complete problem to π; this may lead to confusion (see Section 4).

We also introduce the classes P^{NP} (also known as Δ_2 in the class hierarchy) and L^{NP} (also denoted by $P^{NP[O(\log n)]}$ or Θ_2), which contain the decision problems which can be solved by applying, with a number of calls which is polynomial (respectively, logarithmic) with respect to the size of the instance, a subprogram able to solve an appropriate problem in *NP* (usually, an *NP*-complete problem); and the class *DP* [23] (or DIF^P [2] or BH_2 [18], [25], ...) as the class of languages (or problems) L such that there are two languages $L_1 \in NP$ and $L_2 \in$ co-*NP* satisfying $L = L_1 \cap L_2$. This class is not to be confused with $NP \cap$ co-*NP* (see the warning in, e.g., [22, p. 412]); actually, *DP* contains $NP \cup$ co-*NP* and is contained in L^{NP}. See Figure 1.

Membership to *P*, *NP*, co-*NP*, *DP*, L^{NP} or P^{NP} gives an upper bound on the complexity of a problem (this problem is not more difficult than ...), whereas a hardness result gives a lower bound (this problem is at least as difficult as ...). Still, such results are conditional in some sense; if

for example *P=NP*, they would lose their interest. But we do not know whether or where the classes of complexity collapse.

We now consider the satisfiability problems defined in Section 1.1.

The problems SAT and 3-SAT are two of the basic and most well-known *NP*-complete problems [4], [8, p. 39, p. 46 and p. 259]. More generally, k-SAT is *NP*-complete for $k \geq 3$ and polynomial for $k = 2$. The problem 1-3-SAT, which is obvioulsy in *NP*, is also *NP*-complete [24, Lemma 3.5], [8, p. 259], [15, Rem. 3].

In [15] we proved the following.

Proposition 3 *[15, Th. 10] For every integer $k \geq 3$, the problems U-SAT, U-k-SAT and U-1-3-SAT are equivalent, up to polynomials.* ◊

Using results from [2] and [22, p. 415], it is then rather simple to obtain the following result.

Corollary 4 *For every integer $k \geq 3$,*

(a) the decision problems U-SAT, U-k-SAT and U-1-3-SAT are NP-hard (and co-NP-hard by Remark 2);

(b) the decision problems U-SAT, U-k-SAT and U-1-3-SAT belong to the class DP. ◊

Remark 5 *It is not known whether these problems are DP-complete. In [22, p. 415], it is said that "U-SAT is not believed to be DP-complete".*

We are now ready to investigate the problems of Vertex Cover and Domination.

2 Vertex Covers

In this section, we are going to describe three polynomial reductions:

U-1-3-SAT $\rightarrow_p$ U-VC and U-1-3-SAT $\rightarrow_p$ U-OVC (Theorem 8),

U-VC $\rightarrow_p$ U-SAT (Theorem 11).

The consequence of these reductions is that U-SAT and U-VC have equivalent complexity, that U-VC and U-OVC are *NP*-hard, and that U-VC belongs to *DP*. We shall also show that U-OVC belongs to the class L^{NP}.

2.1 Preliminary Results

The following result characterizes the vertices belonging to at least one optimal VC, through the comparison of two vertex cover numbers, and will be used in the (constructive) proof of Proposition 13.

Lemma 6 *Let $G = (V, E)$ be a graph. For a given vertex $\alpha \in V$, we consider the following graph: $G_\alpha = (V_\alpha, E_\alpha)$, with*

$$V_\alpha = V \cup \{\beta_1, \beta_2\},\ E_\alpha = E \cup \{\alpha\beta_1, \alpha\beta_2\},$$

where, for $1 \leq i \leq 2$, $\beta_i \notin V$. Then α belongs to at least one optimal vertex cover in G if and only if $\phi(G) = \phi(G_\alpha)$.

Proof. (a) Let α be a vertex belonging to at least one optimal vertex cover C in G; then C is a VC in G_α as well, and $\phi(G_\alpha) \leq |C| = \phi(G)$.

On the other hand, let C^* be an optimal VC in G_α. Then obviously $\alpha \in C^*$ and none of the β_i's belongs to C^*. Consequently, $C^* \subseteq V$, C^* is a VC in G, and $\phi(G) \leq |C^*| = \phi(G_\alpha)$.

Therefore, with this assumption on α, we have: $\phi(G) = \phi(G_\alpha)$.

(b) Conversely, assume that $\phi(G) = \phi(G_\alpha)$ for a vertex $\alpha \in V$. Let C^* be an optimal VC in G_α; again, $\alpha \in C^*$, and none of the β_i's belongs to C^*. Then C^* is a VC in G, it has size $\phi(G_\alpha) = \phi(G)$, and it contains α. ◇

Proposition 7 *[19], [8, p. 46 and p. 190] The decision problem VC is NP-complete.* ◇

Actually, we shall only use the fact that the problem VC belongs to NP, in the proofs of Propositions 13 and 14.

2.2 Uniqueness of Vertex Cover

2.2.1 From U-1-3-SAT to U-VC and U-OVC

Theorem 8 *There exists a polynomial reduction from U-1-3-SAT to U-VC and to U-OVC: U-1-3-SAT $\rightarrow_p$ U-VC and U-1-3-SAT $\rightarrow_p$ U-OVC.*

Proof. Going deeper into the proof of the *NP*-completeness of the problem Vertex Cover (see [19], [8, pp. 54–56]), which uses a polynomial reduction from 3-SAT and obviously does not convey the uniqueness of the solution, we describe a polynomial reduction from the problem U-1-3-SAT to U-VC and U-OVC, see Figure 2. If the instance of U-1-3-SAT consists of a set C of m clauses over n variables, we construct

- one vertex denoted by X;
- for each clause c_j, a triangle $T_j = \{a_j, b_j, d_j\}$;
- for each variable x_i, a component $G_i = (V_i = \{x_i, \overline{x}_i\}, E_i = \{x_i\overline{x}_i\})$ and an auxiliary triangle $T_i^* = \{a_i^*, b_i^*, d_i^*\}$ whose vertices are linked to x_i, $\overline{x}_i$ and X by the three edges $x_i a_i^*$, $\overline{x}_i d_i^*$ and $b_i^* X$, called "auxiliary membership edges".

Then we link the components G_i on the one hand, and the triangles T_j on the other hand, according to which literals appear in which clauses

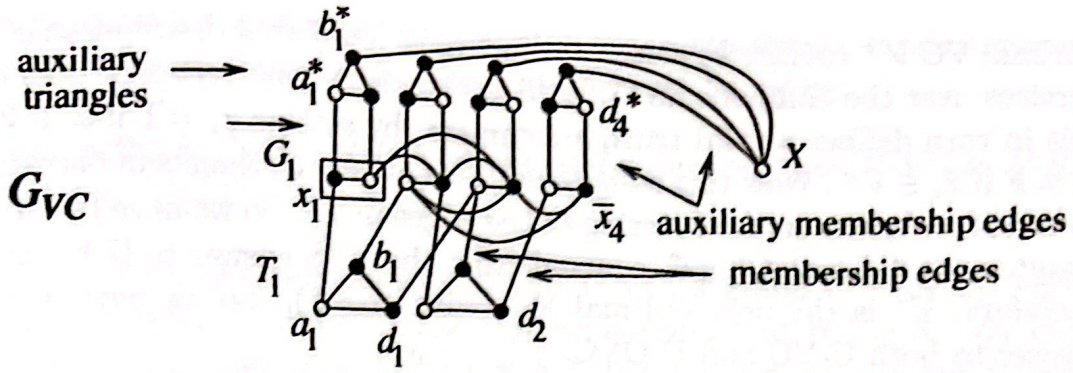

Figure 2: Illustration of the graph constructed for the reduction from U-1-3-SAT to U-OVC, with four variables and two clauses, $c_1 = \{x_1, x_2, x_3\}$, $c_2 = \{\bar{x}_2, x_3, x_4\}$. The sixteen black vertices form the (not unique) optimal vertex cover V^* corresponding to the (not unique) truth assignment $x_1 =$ T, $x_2 =$F, $x_3 =$F, $x_4 =$F 1-3-satisfying the clauses. As soon as we set $V^* \cap (V_1 \cup V_2 \cup V_3 \cup V_4) = \{x_1, \bar{x}_2, \bar{x}_3, \bar{x}_4\}$, the other vertices in V^* are forced.

("membership edges"): for each clause $c_j = \{\ell_1, \ell_2, \ell_3\}$, we link ℓ_1 and a_j, ℓ_2 and b_j, and ℓ_3 and d_j. We also add the triangular set of edges $E'_j = \{\bar{\ell}_1\bar{\ell}_2, \bar{\ell}_1\bar{\ell}_3, \bar{\ell}_2\bar{\ell}_3\}$.

The graph G thus constructed constitutes the instance of U-OVC, and, together with the integer $k = 2m + 3n$, the instance of U-VC. The order of G is $1 + 3m + 5n$.

Note that if V^* is a VC, then each triangle T_j and each auxiliary triangle T_i^* contain at least two vertices, each component G_i at least one vertex, and $|V^*| \geq 2(m+n) + n = 2m + 3n = k$; if $|V^*| = k$, then V^* is optimal, and each triangle contains exactly two vertices, each component G_i exactly one vertex. We can also observe that, because of the edge sets E'_j, at least two vertices among $\bar{\ell}_1, \bar{\ell}_2, \bar{\ell}_3$ belong to any VC.

(a) Let us first assume that the answer to U-1-3-SAT is YES: there is a unique truth assignment 1-3-satisfying the clauses of $\mathcal{C}$. Then, by taking, in each G_i, the vertex corresponding to the literal which is TRUE, in every triangle T_j, the two vertices which are linked to the two false literals of c_j, and in every auxiliary triangle the vertex linked to X and the one linked to the false literal, we obtain a VC V^* whose size is equal to $k = 2m + 3n$, which is optimal. Note that $X \notin V^*$; in fact, once we have put the n vertices corresponding to the true literals in the VC V^* in construction, we have *no choice* for the optimal (up to k) completion of V^*: when we take two vertices in T_j, we *must* take the two vertices which cover the membership edges linked to the two false literals (in Figure 2, the vertices b_1, d_1 and b_2, d_2); similarly, we have no choice either for the auxiliary triangles. So, if another

optimal VC V^+ (of size k) exists, it must have a different distribution of its vertices over the components G_i, still with exactly one vertex in each G_i; this in turn defines a valid truth assignment, by setting $x_i = \mathrm{T}$ if $x_i \in V^+$, $x_i = \mathrm{F}$ if $\overline{x}_i \in V^+$. Now this assignment 1-3-satisfies $\mathcal{C}$, thanks in particular to our observation on the covering of the edges in E'_j. So we have two truth assignments 1-3-satisfying $\mathcal{C}$, contradicting the YES answer to U-1-3-SAT; therefore, V^* is the only optimal VC (with size k). So we have a YES answer to both U-VC and U-OVC.

(b) Assume next that the answer to U-1-3-SAT is NO: this may be either because no truth assignment 1-3-satisfies the instance, or because at least two assignments do; in the latter case, this would lead, using the same argument as in the previous paragraph, to at least two optimal VC (of size $k = 2m + 3n$), and a NO answer to both U-VC and U-OVC. So we are left with the case when the set of clauses $\mathcal{C}$ cannot be 1-3-satisfied. As seen previously when discussing V^+, this implies that no VC of size k exists; this is sufficient to show that the answer is also NO for U-VC.

Assume then that V^* is an optimal VC, of unknown size $|V^*| > 2m+3n$. Then (i) at least one triangle contains three vertices of V^*, or (ii) at least one component G_i contains two vertices of V^*, or (iii) $X \in V^*$. Assume first that one triangle T_j or one auxiliary triangle T_i^* contains three vertices belonging to V^*; this can happen only if the three membership edges, auxiliary or not, starting from this triangle are not covered by their other ends (otherwise, we could save one vertex in the triangle). But then, exchanging in V^* one vertex of this triangle with the other end of its membership edge gives another optimal VC, and a NO answer to U-OVC. So we may assume from now on that all the triangles have exactly two vertices in V^*. Assume next that a component G_i has two vertices belonging to V^*; then inside its auxiliary triangle T_i^*, we have at least two possibilities for choosing the two vertices belonging to V^*, and, once again, a NO answer to U-OVC. So we are left with the case when $X \in V^*$, but again, using the same type of argument, there is choice inside the auxiliary triangles. So in all cases, we have a NO answer to U-OVC. ◊

Corollary 9 *The decision problems UVC and U-OVC are NP-hard.*

Proof. Use Theorem 8 and Corollary 4(a). ◊

2.2.2 An Upper Bound for the Complexity of U-VC

Remark 10 *The method carried out in the proof of the following theorem is quite general and can be used with other types of problems, e.g., those involving the existence of a vertex set with bounded size in a graph: roughly speaking, the clauses constructed below in (a) "describe" the problem, those in (b) deal with the size of the set, and finally the clauses in (c) rule out*

multiple solutions obtained by permutations, symmetries, ..., and guarantee the uniqueness. See also the proof of Theorem 24 below.

Theorem 11 *There exists a polynomial reduction from U-VC to U-SAT: U-VC* $\rightarrow_p$ *U-SAT.*

Proof. We start from an instance of U-VC, a graph $G = (V, E)$ and an integer k, with $V = \{x^1, \ldots, x^{|V|}\}$; we assume that $|V| \geq 3$. We create the set of variables $\mathcal{X} = \{x_i^h : 1 \leq h \leq |V|, 1 \leq i \leq k\}$ and the following clauses:

(a) for each edge $x^h x^\ell \in E$, clauses of size $2k$: $\{x_1^h, x_2^h, \ldots, x_k^h, x_1^\ell, \ldots, x_k^\ell\}$;
(b1) for $1 \leq i \leq k$ and $1 \leq h < \ell \leq |V|$, clauses of size two: $\{\overline{x}_i^h, \overline{x}_i^\ell\}$;
(b2) for $1 \leq i < j \leq k$ and $1 \leq h \leq |V|$, clauses of size two: $\{\overline{x}_i^h, \overline{x}_j^h\}$;
(c) for $1 \leq i < k$ and $1 < \ell \leq |V|$, for $1 \leq h < \ell$ and $i < j \leq k$, clauses of size two: $\{\overline{x}_i^\ell, \overline{x}_j^h\}$.

Note that the number of variables and clauses is polynomial with respect to the order of G, since we may assume that $k \leq |V|$.

Assume that we have a unique VC of size k in G, $V^* = \{x^{p_1}, x^{p_2}, \ldots, x^{p_k}\}$, with $p_1 < p_2 < \ldots < p_k$. Observe that V^* is optimal (otherwise, any optimal VC destroys the uniqueness assumption). Define the assignment $\mathcal{A}_1$ by $\mathcal{A}_1(x_q^{p_q}) = \mathrm{T}$ for $1 \leq q \leq k$, and all the other variables are set FALSE by $\mathcal{A}_1$. This assignment satisfies all the clauses; indeed:

(a) at least one of x^h, x^ℓ belongs to V^*; if, say, $x^h = x^{p_q} \in V^*$, then by definition x_q^h is set TRUE by $\mathcal{A}_1$, and satisfies the clause;

(b1) if $\{\overline{x}_i^h, \overline{x}_i^\ell\}$ is not satisfied for some h, i, ℓ, then $\mathcal{A}_1(x_i^h) = \mathcal{A}_1(x_i^\ell) = \mathrm{T}$, which would mean that two different vertices are the i-th element in V^*;

(b2) if $\{\overline{x}_i^h, \overline{x}_j^h\}$ is not satisfied, this means that x^h appears more than once in V^*;

(c) if $\{\overline{x}_i^\ell, \overline{x}_j^h\}$ is not satisfied for some i, ℓ, with $h < \ell$ and $i < j$, then $\mathcal{A}_1(x_i^\ell) = \mathcal{A}_1(x_j^h) = \mathrm{T}$. This means that $x^\ell = x^{p_i}$ and $x^h = x^{p_j}$; so $\ell = p_i$, $h = p_j$. Now $h < \ell$ implies that $p_j < p_i$, but $i < j$ implies that $p_i < p_j$, a contradiction.

Is $\mathcal{A}_1$ unique? Assume on the contrary that another assignment, $\mathcal{A}_2$, also satisfies the constructed instance of U-SAT. By (a), at least one variable x_i^h or x_j^ℓ is set TRUE by $\mathcal{A}_2$, for every h, ℓ corresponding to an edge $x^h x^\ell$, and for some i or j; so if V^+ is a vertex set which contains the vertex x^h as soon as some variable x_i^h is set TRUE by $\mathcal{A}_2$, then V^+ is a vertex cover. By (b1), for each $i \in \{1, \ldots, k\}$ there is at most one variable with subscript i set TRUE by $\mathcal{A}_2$; this tells us that we have constructed a VC with (at most) k elements. Since such a VC is unique by assumption, we can see that $\mathcal{A}_1$ and $\mathcal{A}_2$ have "selected" the same k vertices, i.e., for each $p_q \in \{p_1, \ldots, p_k\}$,

there is exactly one variable, $x_q^{p_q}$, set TRUE by $\mathcal{A}_1$, and, thanks to (b2), exactly one variable, say $x_s^{p_q}$, set TRUE by $\mathcal{A}_2$. All the other variables with superscript p_q are FALSE by $\mathcal{A}_1$ or $\mathcal{A}_2$. Then, using (c), we can see that necessarily $q = s$ for every p_q, and that $\mathcal{A}_1$ and $\mathcal{A}_2$ must coincide: indeed, assume on the contrary that for some $q \in \{1, \ldots, k\}$, we have $q \neq s$; we treat the case $1 \leq q < s \leq k$, the case $1 \leq s < q \leq k$ being similar. Then if we consider the subscripts smaller than s, there must be one, say v, such that there is a superscript $p_u > p_q$ verifying $\mathcal{A}_2(x_v^{p_u}) = \mathrm{T}$. Now the clause from (c) $\{\overline{x}_v^{p_u}, \overline{x}_s^{p_q}\}$ is not satisfied by $\mathcal{A}_2$, a contradiction.

So a YES answer for U-VC leads to a YES answer for U-SAT. Assume now that the answer to U-VC is negative. If it is negative because there are at least two VC of size k, then we have at least two assignments satisfying the instance of U-SAT: we have seen above how to construct a suitable assignment from a VC, and different VC obviously lead to different assignments. If there is no VC of size k, then there is no assignment satisfying U-SAT, because such an assignment would give a VC of size k, as we have seen above with $\mathcal{A}_2$. So in both cases, a NO answer to U-VC implies a NO answer to U-SAT. ◊

Theorem 12 *For every integer $k \geq 3$, the problems U-SAT, U-1-3-SAT, U-k-SAT and U-VC have equivalent complexity, up to polynomials.*

As a consequence, U-VC belongs to the class DP.

Proof. Simply gather Proposition 3, Theorems 8 and 11, and Corollary 4(b). ◊

Note that it could have been shown directly that U-VC belongs to *DP*.

2.2.3 Two Upper Bounds for the Complexity of U-OVC

We give a first upper bound on the complexity of U-OVC, because the proof is interesting in itself, because it uses a constructive argument (if there is a unique optimal vertex cover, the algorithm can output it), and because for some problems it is sometimes the only available method and result. In the case of Vertex Cover however, we can improve on Proposition 13, and instead of calling a polynomial number of times an algorithm solving a problem in *NP*, we need to call it only a logarithmic number of times, see Proposition 14.

Proposition 13 *The decision problem U-OVC belongs to the class P^{NP}. In case of a YES answer, one can give the only optimal vertex cover within the same complexity.*

Proof. Let $\mathcal{A}_1$ be an algorithm solving the problem VC, which is in *NP*, cf. Proposition 7; using a standard dichotomous process, we obtain an

algorithm $\mathcal{A}_2$ outputting $\phi(G)$ for any graph G, with a logarithmic number of calls to $\mathcal{A}_1$.

Let $G = (V, E)$ be any instance of U-OVC, with n vertices. Run $\mathcal{A}_2$ for G, then, for each vertex $v \in V$, run $\mathcal{A}_2$ for G_v, the graph defined in Lemma 6. By the same lemma, we know, by comparing $\phi(G)$ and $\phi(G_v)$, whether v belongs to at least one optimal VC in G or not. Let $Y = \{v_1, \ldots, v_\ell\}$ be the set of vertices with a positive answer; then necessarily $\ell \in \{\phi(G), \phi(G)+1, \ldots, n\}$. Now if $\ell > \phi(G)$, there is more than one optimal VC in G, whereas if $\ell = \phi(G)$, there is only one, namely, Y.

This shows that we can obtain the answer to U-OVC with $n+1$ calls to $\mathcal{A}_2$, which leads to a polynomial number of calls to an algorithm solving the problem VC, together with negligible operations such as constructing G_v. Moreover, we can construct the optimal VC when there is one. ◊

Proposition 14 *The decision problem U-OVC belongs to the class* L^{NP}.

Proof. We use the same algorithm $\mathcal{A}_2$ as in the proof of the previous proposition, for any graph G. Then, once we have $\phi(G)$, we run an algorithm solving U-VC, for the instance consisting of G and $\phi(G)$. The answer is YES if and only if there is a unique optimal VC in G. Since the problem VC is in *NP* and U-VC is in *DP* (but actually, membership to L^{NP} would suffice), this amounts to a logarithmic number of calls to an algorithm solving a problem in *NP*, plus one call for a problem in *DP*. Because $DP \subseteq L^{NP}$, we are done. ◊

Note that using an algorithm for U-VC without knowing $\phi(G)$ would lead nowhere, because a NO answer cannot be interpreted: either $k < \phi(G)$ and there is no VC, or $k \geq \phi(G)$, but there is more than one VC.

2.3 Related Results: Cliques and Independent Sets

Let $G = (V, E)$ be a graph, and $G^c = (V, E^c)$ be its *complement*: $E^c = \{uv : u \in V, v \in V, u \neq v, uv \notin E\}$. Then the following three statements are equivalent: (a) V^* is a vertex cover in G; (b) $V \setminus V^*$ is an independent set in G; (c) $V \setminus V^*$ induces a clique in G^c. These relationships are simple enough to make it trivial to polynomially transform the problem VC to any one of the following two problems, and *vice-versa*:

Instance: A graph G and an integer k.

Question: Does G admit

(1) an independent set of size at least k? **(2)** a clique of size at least k?

It follows that

– these two problems have the same complexity as the problem VC;

– the problems of a unique clique (with bounded size or optimal) and of a unique independent set (with bounded size or optimal) have the same complexity as U-VC and U-OVC, respectively.

3 Dominating Codes

The approach for dominating codes is quite similar to the one for vertex covers, but slightly more complicated because we shall have to study successively 1-domination, then r-domination for general r.

After some necessary preliminary results, we are going to prove, for $r \geq 1$, the polynomial reductions

U-3-SAT $\rightarrow_p$ U-DC$_1$ and U-3-SAT $\rightarrow_p$ U-ODC$_1$ (Theorem 18),
U-DC$_1$ $\rightarrow_p$ U-DC$_r$ and U-ODC$_1$ $\rightarrow_p$ U-ODC$_r$ (Theorem 20),
U-DC$_r$ $\rightarrow_p$ U-DC$_1$ and U-ODC$_r$ $\rightarrow_p$ U-ODC$_1$ (Proposition 22),
U-DC$_1$ $\rightarrow_p$ U-SAT (Theorem 24).

The consequence of these reductions is that U-DC$_r$ and U-ODC$_r$ are *NP*-hard. Also, U-SAT and U-DC$_r$ have equivalent complexity; as a result, U-DC$_r$ belongs to *DP*. We shall also show that U-ODC$_r$ belongs to the class L^{NP}.

3.1 Preliminary Results

The following lemma will be used in the proof of Theorem 20.

Lemma 15 *Let $r \geq 1$ be any integer and let $G^*_{uv} = (V^*_{uv}, E^*_{uv})$ be the graph defined as follows:*

$$V^*_{uv} = \{u, v\} \cup \{\alpha_i : 1 \leq i \leq r-1\} \cup \{\beta_{i,j} : 1 \leq i \leq r-1, 1 \leq j \leq 3r\},$$

$$E^*_{uv} = \{u\alpha_1, \alpha_1\alpha_2, \ldots, \alpha_{r-1}v\} \cup$$
$$\cup \{\alpha_i\beta_{i,1}, \beta_{i,1}\beta_{i,2}, \ldots, \beta_{i,r}\beta_{i,r+1}, \ldots, \beta_{i,2r-1}\beta_{i,2r} : 1 \leq i \leq r-1\} \cup$$
$$\cup \{\beta_{i,r}\beta_{i,2r+1}, \beta_{i,2r+1}\beta_{i,2r+2}, \ldots, \beta_{i,3r-1}\beta_{i,3r} : 1 \leq i \leq r-1\},$$

see Figure 4 further down. Then $\gamma_r(G^*_{uv}) = r$, $C_1 = \{u\} \cup \{\beta_{i,r} : 1 \leq i \leq r-1\}$ and $C_2 = \{v\} \cup \{\beta_{i,r} : 1 \leq i \leq r-1\}$ are two optimal r-dominating codes in G^*_{uv}, and any optimal r-dominating codes in G^*_{uv} contains $W = \{\beta_{i,r} : 1 \leq i \leq r-1\}$.

Proof. Because the $r-1$ vertices $\beta_{i,2r}$ must be r-dominated by some codeword, at least $r-1$ codewords are necessary. But no vertex can simultaneously r-dominate $\beta_{i,2r}$ and u or v, so at least one more codeword is required, and $\gamma_r(G^*_{uv}) \geq r$. On the other hand, it is quite straightforward to check that C_1 and C_2 are r-dominating codes, of size r, so that they

are optimal, and $\gamma_r(G^*_{uv}) = r$. Finally, for a given $i \in \{1, \ldots, r-1\}$, only $\beta_{i,r}$ can r-dominate both $\beta_{i,2r}$ and $\beta_{i,3r}$, and so $\beta_{i,r}$ belongs to any optimal r-dominating code. ◊

Also note that, for any given i with $1 \leq i \leq r-1$, the vertex $\beta_{i,r}$ r-dominates *exactly* α_i and $\beta_{i,j}$, for $1 \leq j \leq 3r$.

The following lemma, which characterizes the vertices belonging to at least one optimal r-dominating code, through the comparison of two r-domination numbers, is very similar to Lemma 6 for vertex covers; it will be used for Proposition 26. It is a simplified version of [14, Cor. 2].

Lemma 16 *Let $G = (V, E)$ be a graph, and let $r \geq 1$ be any integer. For a given vertex $\alpha \in V$, we consider the following graph: $G_\alpha = (V_\alpha, E_\alpha)$, with*

$$V_\alpha = V \cup \{\beta_i : 1 \leq i \leq r\},\ E_\alpha = E \cup \{\alpha\beta_1\} \cup \{\beta_i\beta_{i+1} : 1 \leq i \leq r-1\},$$

where for $i \in \{1, \ldots, r\}$, $\beta_i \notin V$. Then α belongs to at least one optimal r-dominating code in G if and only if $\gamma_r(G) = \gamma_r(G_\alpha)$. ◊

Proposition 17 *[8, p. 75 and p. 190, for $r = 1$], [14, Prop. 9] Let $r \geq 1$ be any integer. The decision problem DC_r is NP-complete.* ◊

Actually, we shall only use the fact that DC_r belongs to *NP*, for Propositions 26 and 27. Note that the proofs of Proposition 17 do not deal with the problem of the uniqueness of a solution.

3.2 Uniqueness of Dominating Code

3.2.1 From U-3-SAT to U-DC$_1$ and U-ODC$_1$

Theorem 18 *There exists a polynomial reduction from U-3-SAT to U-DC$_1$ and to U-ODC$_1$: U-3-SAT $\rightarrow_p$ U-DC$_1$ and U-3-SAT $\rightarrow_p$ U-ODC$_1$.*

Proof. The construction of a graph is common to the two reductions, then we add an integer k for U-DC$_1$. We start from an instance of U-3-SAT, a collection $\mathcal{C}$ of m clauses over a set $\mathcal{X}$ of n variables. For each variable $x_i \in \mathcal{X}$, $1 \leq i \leq n$, we construct the graph $G_i = (V_i, E_i)$ as follows (see Figure 3):

$$V_i = \{x_i, \overline{x}_i, a_i, b_i\} \cup \{\alpha_{i,\ell} : 1 \leq \ell \leq 3\},$$

$$E_i = \{x_i a_i, \overline{x}_i a_i, a_i b_i\} \cup \{x_i \alpha_{i,\ell}, \overline{x}_i \alpha_{i,\ell} : 1 \leq \ell \leq 3\}.$$

Then for each clause c_j, $1 \leq j \leq m$, containing three literals, we create one vertex, A_j, and link it to the three vertices corresponding, in the graphs G_i, to the literals of c_j. This is our graph G; its order is $7n + m$. Additionally, we set $k = 2n$ for U-DC$_1$.

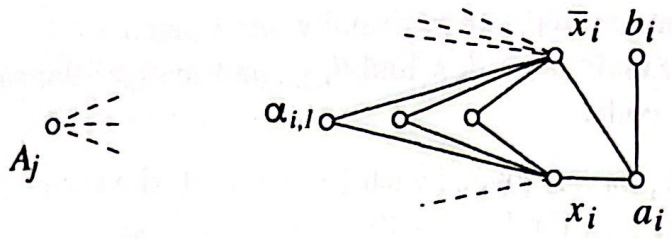

Figure 3: The graph G constructed in the proof of Theorem 18.

Note already that, because of the vertices b_i and $\alpha_{i,\ell}$, any optimal 1-dominating code in G contains x_i or $\overline{x}_i$, and a_i or b_i, for all $i \in \{1, \ldots, n\}$. Consequently, $\gamma_1(G) \geq 2n = k$.

We claim that there is a unique solution to 3-SAT if and only if there is a unique optimal 1-dominating code, and if and only if there is a unique 1-dominating code of size (at most) k, in G.

(1) Assume first that there is a unique truth assignment satisfying all the clauses. We construct the following code C: for $i \in \{1, \ldots, n\}$, among the vertices $x_i \in V_i$, $\overline{x}_i \in V_i$, we put in C the vertex x_i if the literal x_i has been set TRUE, the vertex $\overline{x}_i$ if the literal x_i is FALSE, and we add a_i. Then C is a 1-dominating code; in particular, every vertex A_j is 1-dominated by at least one codeword since every clause contains at least one true literal. The code C has size $k = 2n$ and is optimal. Moreover, once x_i or $\overline{x}_i$ is a codeword, the *only way* to complete the code with not more than n additional codewords is to take a_i, so that $\overline{x}_i$ or x_i is 1-dominated by C, since no A_j is a codeword. The code C is unique: suppose on the contrary that C^* is another 1-dominating code of size $2n$ in G. Then $|C^* \cap V_i| = 2$ for all $i \in \{1, \ldots, n\}$, no A_j is a codeword, and exactly one of x_i and $\overline{x}_i$ is in C^*. This defines a valid truth assignment for $\mathcal{X}$, by setting $x_i = \mathrm{T}$ if $x_i \in C^*$, $x_i = \mathrm{F}$ if $\overline{x}_i \in C^*$. Since $C \neq C^*$, this assignment is different from the assignment used to build C. But the fact that C^* 1-dominates A_j for all j shows that there is a codeword x_i or $\overline{x}_i$ 1-dominating A_j, which means that the clause c_j is satisfied. Therefore, we have a second assignment satisfying the instance of 3-SAT, a contradiction. We can conclude that both problems, U-DC$_1$ and U-ODC$_1$, also have a YES answer.

(2) Assume next that the answer to U-3-SAT is NO: this may be either because no truth assignment satisfies the instance, or because at least two assignments do; in the latter case, this would lead, using the same argument as previously, to at least two optimal 1-dominating codes of size $k = 2n$, and a NO answer to U-DC$_1$ and to U-ODC$_1$. So we are left with the case when the set of clauses $\mathcal{C}$ cannot be satisfied. This implies that no 1-dominating code of size $k = 2n$ exists, as seen with C^*; this is sufficient

to end the proof for U-DC$_1$, but we have to go on for U-ODC$_1$: assume then that C is an optimal 1-dominating code of unknown size $|C| > 2n$. We know that each V_i contains at least two codewords. Where can the extra codeword(s) be?

Suppose that a vertex A_{j_0} is a codeword. If the three vertices in $\{x_i, \overline{x}_i : 1 \leq i \leq n\}$ to which A_{j_0} is linked are codewords, then A_{j_0} could have been saved. So there is at least one neighbour of A_{j_0}, say x_{i_0}, which does not belong to C. Now $\overline{x}_{i_0} \in C$, x_{i_0} is 1-dominated by $A_{j_0} \in C$, and either $C \cap V_{i_0} = \{\overline{x}_{i_0}, a_{i_0}\}$ or $C \cap V_{i_0} = \{\overline{x}_{i_0}, b_{i_0}\}$ may be part of an optimal solution, i.e., we have a NO answer for U-ODC$_1$.

So we can assume from now on that no A_j is a codeword, and that there is at least one set V_{i_0} containing at least three codewords. If it is more than three, then codewords can be spared. If it is exactly three, then x_{i_0} and $\overline{x}_{i_0}$ are codewords, so that some vertices A_j linked to them are 1-dominated by C thanks to x_{i_0} and $\overline{x}_{i_0}$: otherwise, two codewords inside V_{i_0} would have been sufficient. Now we have a choice for the third codeword in V_{i_0}: it can be either a_{i_0} or b_{i_0}.

In all cases, we have proved that there can be several optimal 1-dominating codes, i.e., we have a NO answer for G, the constructed instance of U-ODC$_1$. ◊

Remark 19 *The fact that all the clauses have degree three has no importance whatsoever, and the proof could also work using any problem U-k-SAT, $k \geq 3$, or U-SAT. Only the degrees of the vertices A_j would be affected.*

3.2.2 Extension to $r \geq 2$

Theorem 20 *Let $r \geq 2$ be any integer. There is a polynomial reduction from U-ODC$_1$ to U-ODC$_r$ and from U-DC$_1$ to U-DC$_r$: U-ODC$_1$ $\rightarrow_p$ U-ODC$_r$ and U-DC$_1$ $\rightarrow_p$ U-DC$_r$.*

Proof. This proof is inspired by the proof of Proposition 9 in [14]. We start from a graph $G = (V, E)$ and an integer k for U-DC$_1$, and the same graph G for U-ODC$_1$.

The graph $G^* = (V^*, E^*)$ is common to the reduction to U-DC$_r$ and to U-ODC$_r$ (see Figure 4), and is defined by setting, for each edge $e = uv \in E$,

$$V_e^* = \{\alpha_{e,i} : 1 \leq i \leq r-1\} \cup \{\beta_{e,i,j} : 1 \leq i \leq r-1, 1 \leq j \leq 3r\},$$

$$E_e^* = \{u\alpha_{e,1}, \alpha_{e,1}\alpha_{e,2}, \ldots, \alpha_{e,r-2}\alpha_{e,r-1}, \alpha_{e,r-1}v\} \cup$$
$$\{\alpha_{e,i}\beta_{e,i,1}, \beta_{e,i,1}\beta_{e,i,2}, \ldots, \beta_{e,i,r}\beta_{e,i,r+1}, \ldots, \beta_{e,i,2r-1}\beta_{e,i,2r} : 1 \leq i \leq r-1\}$$
$$\cup \{\beta_{e,i,r}\beta_{e,i,2r+1}, \beta_{e,i,2r+1}\beta_{e,i,2r+2}, \ldots, \beta_{e,i,3r-1}\beta_{e,i,3r} : 1 \leq i \leq r-1\}.$$

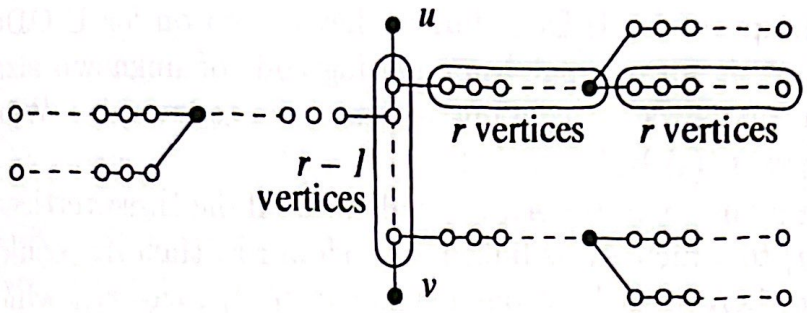

Figure 4: How the edge $e = uv \in E$ gives V_e^* and E_e^*. The black vertices on the branches are the vertices $\beta_{e,i,r}$.

Then we set

$$V^* = V \cup (\cup_{e \in E} V_e^*),\ E^* = \cup_{e \in E} E_e^*.$$

Finally, for U-DC$_r$, we put $k^* = k + (r-1)|E|$. The order of G^* is $|E|(r-1)(3r+1)$.

(1) We claim that an instance of U-ODC$_1$ is positive if and only if the corresponding instance of U-ODC$_r$ is.

(a) First, we assume that there is a YES answer in G for U-ODC$_1$: there is a unique optimal 1-dominating code C in G. Let W be the set consisting of the $(r-1)|E|$ vertices $\beta_{e,i,r}$, $e \in E$, $1 \leq i \leq r-1$. Note that W r-dominates exactly $V^* \setminus V$, and that, by Lemma 15, any optimal r-dominating code in G^* contains W. Because C is 1-dominating in G, clearly $C^* = C \cup W$ is r-dominating in G^*. Note in particular that two vertices u and v at distance 1 in G are at distance r in G^*.

Because moreover C is optimal, the code C^* is also optimal: assume on the contrary that C^+ is an optimal r-dominating code in G^*, with $|C^+| < |C^*|$. We proceed as in the proof of Proposition 9 in [14]: the subcode $C^+ \setminus W$ must r-dominate all the vertices in V, and if a codeword in $V^* \setminus V$ performs part of this task, it can be replaced by a vertex in V. After such replacements have possibly been made, we have a code $C^\times$ such that $C^\times \cap V$ r-dominates, in G^*, all the vertices in V; this implies that in G, $C^\times \cap V$ 1-dominates all the vertices. But $|C^\times \cap V| \leq |C^+ \setminus W| < |C^* \setminus W| = |C| = \gamma_1(G)$, i.e., $|C^\times \cap V| < \gamma_1(G)$, which is impossible.

Finally, because we assumed that C was the only optimal 1-dominating code in G, the code $C^* = C \cup W$ is the only optimal r-dominating code in G^*. Suppose on the contrary that C^+ is another optimal code in G^*: $|C^+| = |C^*| = |C| + |W|$.

(i) $C^+ \cap V = C^+ \setminus W$, or, equivalently, $(V^* \setminus V) \cap C^+ = W$. Then, as we have already seen, $C^+ \cap V$ is 1-dominating in G, and either $C^+ \cap V = C^* \cap V$, which leads to $C^+ = C^*$, or $C^+ \cap V \neq C^* \cap V$, in which case we have two

optimal 1-dominating codes in G, $C^+ \setminus W$ and $C^* \setminus W = C$. In both cases, we have a contradiction.

(ii) $C^+ \cap V \neq C^+ \setminus W$. Then there is at least one codeword $z \in C^+$ belonging to some $V_e^* \setminus \{\beta_{e,i,r} : 1 \leq i \leq r-1\}$, with $e = uv \in E$. Then z must r-dominate u or v or both; otherwise, z is useless and can be saved. For the same reason, $u \notin C^+$ and $v \notin C^+$; but then $(C^+ \setminus \{z\}) \cup \{u\}$ and $(C^+ \setminus \{z\}) \cup \{v\}$ are also both r-dominating in G^*. By similarly replacing all the t codewords in $(V^* \setminus W) \setminus V$, $t \geq 1$, by codewords in V, we have 2^t codes included in $V \cup W$ which are all r-dominating in G^*, which implies that their intersections with V all are 1-dominating in G; moreover, these intersections have size (at most) $|C^+| - |W| = |C|$, i.e., are optimal. Finally, there are at least two of them, so at least one is different from C, contradicting its uniqueness.

This shows that a YES answer to U-ODC$_1$ leads to a YES answer to U-ODC$_r$.

(b) And a NO answer to U-ODC$_1$ leads to a NO answer to U-ODC$_r$, since two optimal 1-dominating codes in G, C_1 and C_2, give two optimal r-dominating codes in G^*, $C_1 \cup W$ and $C_2 \cup W$.

This ends the part for U-ODC$_r$.

(2) We claim that an instance of U-DC$_1$ is positive if and only if the corresponding instance of U-DC$_r$ is.

(a) First, we assume that there is a YES answer for U-DC$_1$: there is a unique 1-dominating code C with size at most k in G. Obviously, C is optimal, otherwise any optimal 1-dominating code in G would contradict the uniqueness of C. Consider the code $C^* = C \cup W$ in G^*, of size $k + (r-1)|E| = k^*$. Exactly as in Step (1) of this proof, C^* is r-dominating, is optimal, and is the only r-dominating code of size at most k^* in G^*. So the answer to U-DC$_r$ is also positive.

(b) Next, we assume that the answer to U-DC$_1$ is NO: either there is no 1-dominating code with size at most k in G, or there is more than one. In the latter case, we have more than one r-dominating code with size at most k^* in G^*: simply add the set W to the codes in G. So we assume that we are in the first case. This implies in particular that $\gamma_1(G) > k$; using the same argument as in Step (1), we can see that $\gamma_r(G^*) > k+(r-1)|E| = k^*$, and there is no r-dominating code with size at most k^* in G^*. In all cases, the answer to U-DC$_r$ is NO. ◊

Corollary 21 *Let $r \geq 1$ be any integer. The decision problems U-DC$_r$ and U-ODC$_r$ are NP-hard.* ◊

Proposition 22 *Let $r \geq 2$ be any integer. There is a polynomial reduction from U-DC$_r$ to U-DC$_1$ and from U-ODC$_r$ to U-ODC$_1$: U-DC$_r$ $\rightarrow_p$ U-DC$_1$ and U-ODC$_r$ $\rightarrow_p$ U-ODC$_1$.*

Proof. Let (G,k) be an instance of U-DC$_r$ and G be an instance of U-ODC$_r$, for $r \geq 2$. The instance for U-DC$_1$ is simply (G^r,k), and G^r for U-ODC$_1$, where G^r is the r-th power of G: obviously, by Lemma 1, there is a unique 1-dominating code of size k in G^r if and only if there is a unique r-dominating code of size k in G, and there is a unique optimal 1-dominating code in G^r if and only if there is a unique optimal r-dominating code in G. ◊

Corollary 23 *Let $r_1 \geq 1$ and $r_2 \geq 1$ be any integers.*

(a) The decision problems U-DC$_{r_1}$ and U-DC$_{r_2}$ are equivalent, up to polynomials.

(b) The decision problems U-ODC$_{r_1}$ and U-ODC$_{r_2}$ are equivalent, up to polynomials. ◊

3.2.3 An Upper Bound for the Complexity of U-DC$_r$

Theorem 24 *There exists a polynomial reduction from U-DC$_1$ to U-SAT: U-DC$_1$ $\rightarrow_p$ U-SAT.*

Proof. We fully use Remark 10: we start from an instance of U-DC$_1$, a graph $G=(V,E)$ and an integer k, with $V=\{x^1,\ldots,x^{|V|}\}$; we assume that $|V| \geq 3$. We create the set of variables $\mathcal{X}=\{x_i^h : 1 \leq h \leq |V|, 1 \leq i \leq k\}$ and the following clauses:

(a) for each vertex $x^h \in V$ with neighbours $x^{h_1},\ldots,x^{h_s}$, clauses of size $(s+1)k$: $\{x_1^h,x_2^h,\ldots,x_k^h,x_1^{h_1},\ldots,x_k^{h_1},\ldots,x_1^{h_s},\ldots,x_k^{h_s}\}$;

(b1) for $1 \leq i \leq k$ and $1 \leq h < \ell \leq |V|$, clauses of size two: $\{\overline{x}_i^h,\overline{x}_i^\ell\}$;

(b2) for $1 \leq i < j \leq k$ and $1 \leq h \leq |V|$, clauses of size two: $\{\overline{x}_i^h,\overline{x}_j^h\}$;

(c) for $1 \leq i < k$ and $1 < \ell \leq |V|$, for $1 \leq h < \ell$ and $i < j \leq k$, clauses of size two: $\{\overline{x}_i^\ell,\overline{x}_j^h\}$.

Compared to the proof of Theorem 11, only the clauses in (a) are different: they convey the fact that among the vertices $x^h,x^{h_1},\ldots,x^{h_s}$, at least one must be put in a 1-dominating code. The proof then goes exactly as for Theorem 11. ◊

By Proposition 22 or its corollary, this immediately implies that there is a polynomial reduction from U-DC$_r$ to U-SAT.

Theorem 25 *Let $r \geq 1$ and $k \geq 3$ be any integers. The problem U-DC$_r$ has complexity equivalent to that of U-SAT, U-k-SAT and U-1-3-SAT.*

As a consequence, U-DC$_r$ belongs to the class DP. ◊

Note that it could have been shown directly that U-DC$_r$ belongs to *DP*.

3.2.4 Two Upper Bounds for the Complexity of U-ODC$_r$

The (constructive) membership of U-ODC$_r$ to the class P^{NP} and the membership to L^{NP} can be proved in exactly the same way as in Section 2.2.3 for U-OVC. This time, it is the characterizing Lemma 16 which must be used, together with Proposition 17.

Proposition 26 *For $r \geq 1$, the decision problem U-ODC$_r$ belongs to the class P^{NP}. In case of a YES answer, one can give the only optimal r-dominating code within the same complexity.* ◊

Proposition 27 *For $r \geq 1$, the decision problem U-ODC$_r$ belongs to L^{NP}.* ◊

4 Conclusion

Corollary 4 states that the three problems, U-SAT, U-1-3-SAT and U-k-SAT ($k \geq 3$), are equivalent and lie somewhere in the vertically hatched area of Figure 5, but probably not in *DP-C*, cf. Remark 5.

Theorems 12 and 25 state that U-VC and U-DC$_r$, $r \geq 1$, have the same complexity as the above three problems, and consequently are located in the same hatched area.

We have also established that the decision problems U-OVC and U-ODC$_r$, $r \geq 1$, belong to the class L^{NP}, see Propositions 14 and 27, and that they are *NP*-hard, see Corollaries 9 and 21. This means that they lie within the areas that are hatched horizontally or vertically. Moreover, all the problems U-ODC$_r$ are equivalent between each other, by Corollary 23(b).

We have the same conclusions for Cliques and Independent Sets (see Section 2.3).

In [2], the authors wonder whether

(A) U-SAT is *NP*-hard, but here we believe that what they mean is: does there exist a *polynomial* reduction from an *NP*-complete problem to U-SAT? i.e., they use the *second* definition of *NP*-hardness;

finally, they show that (A) is true if and only if

(B) U-SAT is *DP*-complete.

So, if one is careless and considers that U-SAT is *NP*-hard without checking according to which definition, one might easily jump too hastily to the conclusion that U-SAT is *DP*-complete, which, to our knowledge, is not known to be true or not. As for U-3-SAT, we do not know where to locate it more precisely either; in [3] the problems U-k-SAT and more particularly U-3-SAT are studied, but it appears that they are versions where the given set of clauses has zero or one solution, which makes quite a difference with our problem.

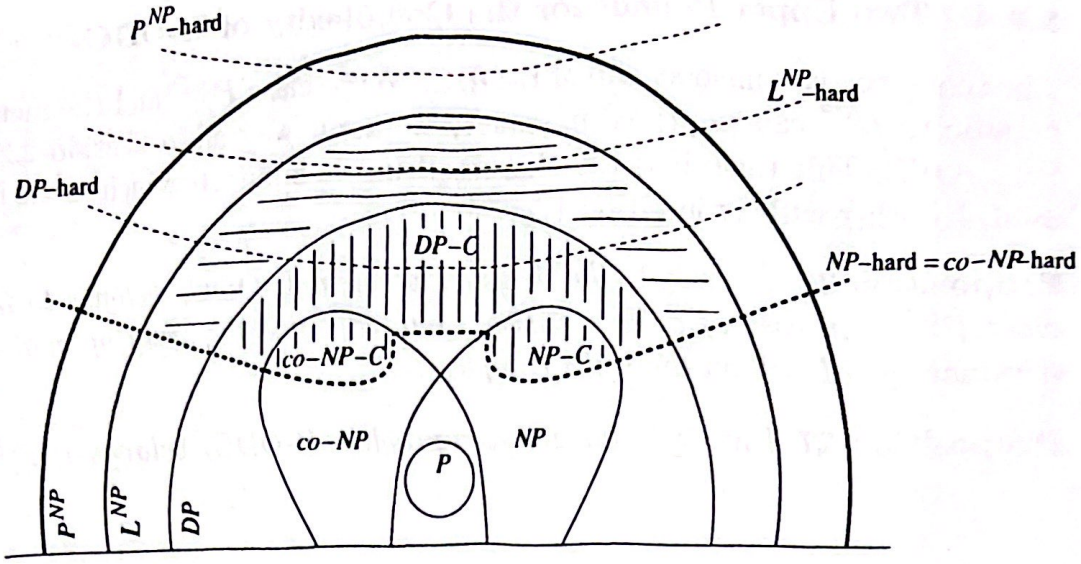

Figure 5: Some classes of complexity: Figure 1 re-visited.

Open problem 1 (general). For $k \geq 3$ and $r \geq 1$, improve the location of U-SAT, U-k-SAT, U-1-3-SAT, U-VC, U-OVC, U-DC$_r$ and U-ODC$_r$, within the classes of complexity.

Open problem 2. It is easy to establish that U-OVC $\rightarrow_p$ U-ODC$_1$ (and U-OVC $\rightarrow_p$ U-ODC$_r$, $r \geq 2$). What more can be said about the relationship between U-OVC and U-ODC$_1$ (and U-ODC$_r$)?

Open problem 3 (conjecture). Under the assumption $P \neq NP$, U-OVC is not equivalent to U-VC, and U-ODC$_r$ is not equivalent to U-DC$_r$, $r \geq 1$.

Finally, in [6] (respectively, [9]), characterizations of the trees (respectively, the block graphs, which are a class of graphs including the trees) admitting a unique optimal 1-dominating code are given. This result, together with a linear algorithm determining optimal 1-dominating codes in block graphs, then allows to show that the sub-problem of U-ODC$_1$ where the instance is any block graph is linear.

Open problem 4. Is it possible to extend this kind of result to any integer $r \geq 1$? to other classes of graphs?

References

[1] J.-P. BARTHÉLEMY, G. D. COHEN and A. C. LOBSTEIN: *Algorithmic Complexity and Communication Problems*, London: University College of London, 1996.

[2] A. BLASS and Y. GUREVICH: On the unique satisfiability problem, *Information and Control*, Vol. 55, pp. 80-88, 1982.

[3] C. CALABRO, R. IMPAGLIAZZO, V. KABANETS and R. PATURI: The complexity of Unique k-SAT: an isolation lemma for k-CNFs, *Journal of Computer and System Sciences*, Vol. 74, pp. 386–393, 2008.

[4] S. A. COOK: The complexity of theorem-proving procedures, *Proceedings of 3rd Annual ACM Symposium on Theory of Computing*, pp. 151–158, 1971.

[5] T. CORMEN: Algorithmic complexity, in: K. H. Rosen (ed.) *Handbook of Discrete and Combinatorial Mathematics*, pp. 1077–1085, Boca Raton: CRC Press, 2000.

[6] M. FISCHERMANN: Block graphs with unique minimum dominating sets, *Discrete Mathematics*, Vol. 240, pp. 247–251, 2001.

[7] M. FRANCES and A. LITMAN: On covering problems of codes, *Theory of Computing Systems*, Vol. 30, No. 2, pp. 113–119, 1997.

[8] M. R. GAREY and D. S. JOHNSON: *Computers and Intractability, a Guide to the Theory of NP-Completeness*, New York: Freeman, 1979.

[9] G. GUNTHER, B. HARTNELL, L. R. MARKUS and D. RALL: Graphs with unique minimum dominating sets, *Congressus Numerantium*, Vol. 101, pp. 55–63, 1994.

[10] T. W. HAYNES, S. T. HEDETNIEMI and P. J. SLATER: *Fundamentals of Domination in Graphs*, New York: Marcel Dekker, 1998.

[11] E. HEMASPAANDRA, H. SPAKOWSKI and J. VOGEL: The complexity of Kemeny elections, *Theoretical Computer Science*, Vol. 349, pp. 382–391, 2005.

[12] L. HEMASPAANDRA: Complexity classes, in: K. H. Rosen (ed.) *Handbook of Discrete and Combinatorial Mathematics*, pp. 1085–1090, Boca Raton: CRC Press, 2000.

[13] I. S. HONKALA and A. C. LOBSTEIN: On the complexity of calculating the minimum norm of a binary code, *Proc. Workshop on Coding and Cryptography '99*, pp. 21–27, Paris, 1999.

[14] O. HUDRY and A. LOBSTEIN: More results on the complexity of domination problems in graphs, *International Journal of Information and Coding Theory*, to appear.

[15] O. HUDRY and A. LOBSTEIN: Some complexity considerations on the uniqueness of solutions for satisfiability and colouring problems, submitted.

[16] O. HUDRY and A. LOBSTEIN: On the complexity of determining whether there is a unique Hamiltonian cycle in a graph, submitted.

[17] O. HUDRY and A. LOBSTEIN: Unique (optimal) solutions: complexity results for identifying and locating-dominating codes, submitted.

[18] D. S. JOHNSON: A catalog of complexity classes, in: *Handbook of Theoretical Computer Science, Vol. A: Algorithms and Complexity*, van Leeuwen, Ed., Chapter 2, Elsevier, 1990.

[19] R. M. KARP: Reducibility among combinatorial problems, in: R. E. Miller and J. W. Thatcher (eds.) *Complexity of Computer Computations*, pp. 85–103, New York: Plenum Press, 1972.

[20] A. McLOUGHLIN: The complexity of computing the covering radius of a code, *IEEE Trans. Inform. Th.*, Vol. 30, pp. 800–804, 1984.

[21] A. MERTZ: On the complexity of multicovering radii, *IEEE Trans. Inform. Th.*, Vol. 50, pp. 1804–1808, 2004.

[22] C. H. PAPADIMITRIOU: *Computational Complexity*, Reading: Addison-Wesley, 1994.

[23] C. H. PAPADIMITRIOU and M. YANNAKAKIS: The complexity of facets (and some facets of complexity), *Journal of Computer and System Sciences*, Vol. 28, pp. 244–259, 1984.

[24] T. J. SCHAEFER: The complexity of satisfiability problems, *Proceedings of 10th Annual ACM Symposium on Theory of Computing*, pp. 216–226, 1978.

[25] https://complexityzoo.uwaterloo.ca/Complexity_Zoo