REVISTA DE LA UNIÓN MATEMÁTICA ARGENTINA
Vol. 68, No. 1, 2025, Pages 187–203
Published online: May 20, 2025
https://doi.org/10.33044/revuma.3349

ON THE MAXIMUM WEIGHTED IRREDUNDANT SET PROBLEM

RICARDO D. KATZ AND DANIEL SEVERÍN

Abstract. We present a generalization of a well-known domination parameter, the *upper irredundance number*, and address its associated optimization problem, namely the *maximum weighted irredundant set* (MWIS) problem, which models some service allocation problems. We establish a polynomial-time reduction to the *maximum weighted stable set* (MWSS) problem that we use to find graph classes for which the MWIS problem is polynomial, among other results. We formalize these results in the proof assistant Coq. This is mainly convenient in the case of some of them due to the structure of their proofs. We also present a heuristic and an integer programming formulation for the MWIS problem and check that the heuristic delivers good quality solutions through experimentation.

1. Introduction

Domination problems are among the most important optimization problems that combinatorial optimization and graph theory address. This is due to at least two reasons. On the one hand, we have their great applicability to real-life problems (see below an example corresponding to the problem considered in this paper). On the other hand, we have the NP-completeness of the basic domination problems and their relationships to other NP-complete problems, and the subsequent interest in finding polynomial-time solutions to domination problems for special graph classes. As a consequence, there is a vast literature about the parameters defining these problems (e.g., we refer the reader to [12, 13] for more information and additional references). Indeed, according to Favaron et al. [8], more than 1500 research papers about dominating sets have been published up to 2002, and this topic is still active nowadays (e.g., see [1]). As we shall see below, this extensive research also includes weighted versions of the main domination parameters. This work concerns a weighted generalization of a parameter introduced in [3], the upper irredundance number.

To precisely introduce the problem considered here and its relationship to the basic domination problems, let $G = (V, E)$ be a simple graph. Given $v \in V$,

2020 *Mathematics Subject Classification.* Primary 05C69; Secondary 03F55, 90C60.
Key words and phrases. maximum weighted irredundant set, polynomial time, heuristic, Coq.
This work was supported by grants PID-UNR ING538 and 80020180100091UR.

we denote by $N(v)$ the *neighborhood* $\{u : (v,u) \in E\}$ of v, and by $N[v]$ its *closed neighborhood* $N(v) \cup \{v\}$. These concepts are extended to sets of vertices as follows: given $S \subseteq V$, $N(S) \doteq \bigcup_{v \in S} N(v)$ and $N[S] \doteq \bigcup_{v \in S} N[v]$. A set $S \subseteq V$ is called *stable* if $N(S) \cap S = \emptyset$. The *independence number* $\alpha(G)$ is the maximum cardinality of a stable set in G. A vertex v is said to *dominate* a vertex u when $u \in N[v]$, or equivalently when $d(v,u) \leq 1$, where $d(v,u)$ is the distance between v and u (i.e., the number of edges in a shortest path connecting them). A set $D \subseteq V$ is called *dominating* if $N[D] = V$. The *domination number* $\gamma(G)$ is the minimum cardinality of a dominating set in G, and the *upper domination number* $\Gamma(G)$ is the maximum cardinality of a minimal dominating set in G. The *independence domination number* $\iota(G)$ is the minimum cardinality of a set which is stable and dominating simultaneously. Given a set $D \subseteq V$ and a vertex $v \in D$, $s_D(v) \doteq N[v] - N[D - \{v\}]$ is the set of *private vertices* of v in D (it contains those vertices only dominated by v). The set D is called *irredundant* if $s_D(v) \neq \emptyset$ for all $v \in D$. In other words, each vertex of D must dominate at least one vertex not dominated by any other vertex of D, see Figure 1 below for an example. The *upper irredundance number* $\mathrm{IR}(G)$ is the maximum cardinality of an irredundant set in G. The *lower irredundance number* $\mathrm{ir}(G)$ is the minimum cardinality of a maximal irredundant set in G.

The Cockayne–Hedetniemi domination chain ([12, Theorem 3.10]) states that $\mathrm{ir}(G) \leq \gamma(G) \leq \iota(G) \leq \alpha(G) \leq \Gamma(G) \leq \mathrm{IR}(G)$ for any graph G. Moreover, this chain of inequalities is still valid for a weighted version of these parameters [22], i.e., when the objective is to maximize/minimize the weight of the considered sets instead of their cardinality. Given a vector of positive weights $w \in \mathbb{Z}_+^V$ (associated with the vertices of G) and a set of vertices $S \subseteq V$, the *weight* of S is defined as $w(S) \doteq \sum_{v \in S} w(v)$. Weighted parameters are denoted with w as a subscript, e.g., $\alpha_w(G)$ stands for the maximum weight of a stable set in G.

The *maximum weighted stable set* problem, corresponding to α_w, and the *minimum weighted dominating set* problem, corresponding to γ_w, are widely studied NP-hard problems. On the other hand, the problems associated with the remaining weighted parameters have recently started to gain attention among researchers. For instance, [21] proposes several approaches to solve (a generalized version of) the *weighted independent dominating set* problem, corresponding to ι_w, and [2] studies the *weighted upper dominating set* (WUDS) problem, where it is proved (among other results) that the computation of the corresponding parameter Γ_w is strongly NP-hard even in a subfamily of subcubic split graphs.

In this work, we study the problem associated with IR_w, that is:

Maximum Weighted Irredundant Set (MWIS) Problem. Given a simple graph $G = (V,E)$ and a weight vector $w \in \mathbb{Z}_+^V$, find an irredundant set D such that $w(D)$ is maximum.

This is the weighted version of the *upper irredundant set* (UIS) problem.

The weighted upper domination parameters Γ_w and IR_w can be used to model certain service allocation problems. Consider a city divided into areas where a service whose quality decreases with the distance to its location (e.g., mobile phone

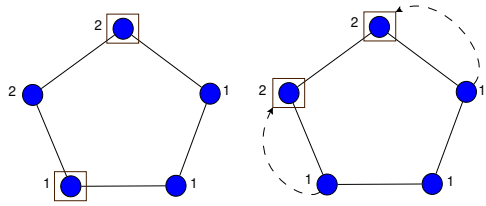

FIGURE 1. Left: a maximum weighted minimal dominating set ($\Gamma_w(G) = 3$). Right: a maximum weighted irredundant set ($\mathrm{IR}_w(G) = 4$), where the dashed arrows link vertices of the irredundant set to their private vertices.

connection) is to be provided. Assume that the service is properly provided to the area in which it is located, as well as to its neighboring areas. If each allocation of the service has a fixed high cost, no allocation to an area is desired if all the areas covered by this allocation are covered by other allocated areas. Thus, if the city is modeled by a graph whose vertices are the areas and there is an edge between two vertices if the corresponding areas are neighbors, then the set of allocated areas must correspond to an irredundant set of this graph. Besides, suppose that each area has a profit associated with it (e.g., representing the proportion of people who would benefit from the allocation of the service to that area), and that it is desired to maximize the sum of these profits, even if this may leave some areas with no (or poor) service. This can be modeled as a MWIS problem where the weights of the vertices are given by the profits of the corresponding areas. If it is required that all the areas were covered, then it becomes a WUDS problem. As an example, consider the graph of Figure 1, which models a city divided into 5 areas. The weights are displayed next to the vertices, so in this example there are two neighbor areas with the highest profit. A solution of the WUDS problem is shown on the left, and one of the MWIS problem on the right. Both sets (the places where the services are located) are displayed by vertices inside boxes. Thus, the total profit in the second case is higher, at the expense of leaving an area with no (or poor) service.

This paper is organized as follows. In Section 2 we study some basic properties of the MWIS problem. Section 3 is devoted to presenting a polynomial-time reduction to the MWSS problem which is used to prove that the MWIS problem can be solved in polynomial time on certain input instances (these results were formalized in the proof assistant Coq). In Section 4 we propose a heuristic and an integer programming formulation for solving the MWIS problem, and perform some computational experiments. In particular, we obtain an optimal solution of an instance that models a service allocation on the city of Buenos Aires. Supplementary material concerning this work, which includes the formalized theory, the code of the heuristic, the implementations of integer programming formulations for the MWIS and WUDS problems, and an appendix with further explanations, is available at github.com/aureus123/graph-theory.

2. Basic properties of the MWIS problem

Theoretical bounds are one of the first features that are proposed when a new parameter is analyzed, as they can in particular be used to check the optimality of a given solution. A natural lower bound for $\mathrm{IR}_w(G)$ is $\Gamma_w(G)$ as we pointed out above. An upper bound for $\mathrm{IR}(G)$ is given in [12, Theorem 3.17]: $\mathrm{IR}(G) \leq |V(G)| - \delta(G)$, where $\delta(G)$ is the minimum degree of G. This bound is tight for *complete multipartite graphs*, i.e., graphs composed of stable sets, with an edge connecting any two vertices of different stable sets, and for *complete split graphs*, i.e., graphs composed of a complete graph and a stable set, with an edge connecting any vertex of the complete graph with any vertex of the stable set. On the other hand, the paw and bull graphs (see Figure 2 below) are examples where this bound is not tight, although it is possible to get a better bound at the expense of a procedure of greater computational complexity. In this sense, we propose two upper bounds for $\mathrm{IR}_w(G)$: $m_1 \doteq w(V) - \delta_w(G)$ and $m_2 \doteq w(V) - \delta'_w(G)$, where

$$\delta_w(G) \doteq \min\{w(N[u] - \{v\}) : v \in V,\, u \in N[v]\}$$

and

$$\delta'_w(G) \doteq \min\{w((N[u_1] \cup N[u_2]) - \{v_1, v_2\}) : v_1, v_2 \in V, \\ u_1 \in N[v_1] - N[v_2],\, u_2 \in N[v_2] - N[v_1]\}. \quad (2.1)$$

Note that m_2 is tight for the paw and bull graphs when $w(v) = 1$ for all $v \in V$.

Lemma 2.1. *For any simple graph $G = (V, E)$ and any weight vector $w \in \mathbb{Z}_+^V$, we have $\mathrm{IR}_w(G) \leq m_1$. Moreover, if G does not have any universal vertex (i.e., a vertex v such that $N[v] = V$), then $\mathrm{IR}_w(G) \leq m_2$.*

Proof. Let D be an irredundant set such that $w(D) = \mathrm{IR}_w(G)$, v a vertex in D, and u a private vertex of v. Then, we have $(N[u] - \{v\}) \cap D = \emptyset$, and so $\mathrm{IR}_w(G) = w(D) \leq w(V - (N[u] - \{v\})) = w(V) - w(N[u] - \{v\}) \leq w(V) - \delta_w(G)$. When G does not have universal vertices, D must contain at least two vertices, say v_1 and v_2 (otherwise, i.e., if D was composed of just one vertex v, we would obtain an irredundant set with a weight greater than $w(D)$ by adding to D any vertex not adjacent to v). Let u_1 and u_2 be private vertices of v_1 and v_2, respectively. Then, we have $(N[u_1] \cup N[u_2] - \{v_1, v_2\}) \cap D = \emptyset$, and so $\mathrm{IR}_w(G) = w(D) \leq w(V) - w(N[u_1] \cup N[u_2] - \{v_1, v_2\}) \leq w(V) - \delta'_w(G)$. □

Regarding complexity, MWIS is an NP-hard problem. In fact, Alice McRae in her Ph.D. thesis [17] showed that the UIS problem is NP-hard even in line graphs of bipartite graphs, or equivalently, in {claw, diamond, odd-hole}-free graphs. There are some results of parameterized complexity too. If the UIS problem is parameterized by the size of the output set, then the resulting problem (i.e., determining whether a graph has an irredundant set of size k where k is the parameter) is W[1]-complete in the W hierarchy (for the definition of this hierarchy and for more details about the theory of fixed parameter tractability, see e.g. [5]), i.e., it has the same hardness than deciding if a given graph contains an independent set of

size k. However, deciding whether a graph $G = (V, E)$ has an irredundant set of size $|V| - k$ is fixed-parameter tractable [7].

On the other hand, we can mention some families of graphs where the MWIS problem can be solved in polynomial time. If G is a complete graph and v is a vertex with highest weight, then $\{v\}$ is an irredundant set of maximum weight, and $\mathrm{IR}_w(G) = w(v)$.

Given two instances $(G_1 = (V_1, E_1), w_1)$ and $(G_2 = (V_2, E_2), w_2)$ of the MWIS problem, let (G^+, w) be the instance that consists of the disjoint union of G_1 and G_2, with weights $w(v) = w_i(v)$ if $v \in V_i$ for $i = 1, 2$. Then, we have:

$$\mathrm{IR}_w(G^+) = \mathrm{IR}_{w_1}(G_1) + \mathrm{IR}_{w_2}(G_2).$$

This property allows us to restrict the resolution of the MWIS problem to connected graphs. Now, let $(G^\vee, w)$ be the instance that consists of the join of G_1 and G_2, i.e., the union of G_1 and G_2 with additional edges that connect every vertex of G_1 to every vertex of G_2, and weights defined as above. If D is an irredundant set of $G^\vee$, then it is straightforward to see that there are three possibilities: (1) $D \subseteq V_1$, (2) $D \subseteq V_2$, or (3) $D = \{v_1, v_2\}$, where v_i is not an universal vertex of G_i for $i = 1, 2$. Therefore, we have:

$$\mathrm{IR}_w(G^\vee) = \max\{\mathrm{IR}_{w_1}(G_1), \mathrm{IR}_{w_2}(G_2), M_1 + M_2\},$$

where $M_i = \max\{w_i(v) : v \in V_i, V_i \nsubseteq N[v]\}$ for $i = 1, 2$.

We recall that cographs are graphs that can be constructed from isolated vertices by disjoint union and join operations, and its recognition can be performed in linear time. Thus, the discussion above leads to the following result:

Lemma 2.2. *The MWIS problem is linear on cographs with any vector of positive weights.*

This linearity result can be extended to graphs of bounded cliquewidth due to *Courcelle's Theorem* and the fact that MWIS can be expressed as a LinEMSOL(τ_1) problem (see [4] for more information on the latter problem).

3. A polynomial-time reduction to the MWSS problem

In this section, we present a polynomial-time reduction from the MWIS problem to the MWSS problem. This transformation is useful for two reasons. First, a MWIS problem can be transformed and then solved with state-of-the-art MWSS solvers, see Section 4. Second, we can find graph classes where the MWIS problem is polynomial via this reduction, e.g., $\{\text{claw}, \text{bull}, P_6, \overline{C_6}\}$-free graphs as we shall see below.

Given a simple graph $G = (V, E)$ and a weight vector $w \in \mathbb{Z}_+^V$, we define the instance $(G' = (V', E'), w')$ of the MWSS problem as follows:

$$V' \doteq \{uv : u \in V,\, d(v, u) \leq 1\},$$
$$E' \doteq \{(uv, zr) : uv, zr \in V',\, uv \neq zr,\, d(v, z) \leq 1 \vee d(r, u) \leq 1\},$$
$$w'(uv) \doteq w(u) \text{ for all } uv \in V'.$$

From now on, we decorate a graph with a *prime* to refer to the graph obtained by applying the transformation above; for example, claw$'$ denotes the graph obtained

from claw by this transformation. The same notation will be applied to the vertex set, edge set and weight vector.

Theorem 3.1. *By defining G' and w' as above, we have* $\mathrm{IR}_w(G) = \alpha_{w'}(G')$.

Proof. To prove this theorem, it suffices to show that for each irredundant set D of G there exists a stable set S of G' such that $w'(S) = w(D)$, and for each stable set S of G' there exists an irredundant set D of G such that $w(D) = w'(S)$.

Let us start with the proof of the first property, so assume D is irredundant. For each $v \in D$, let p_v be a private vertex of v in D (such a vertex exists since D is irredundant). Note that we have $vp_v \in V'$ since $p_v \in N[v]$. Thus, taking S as the set $\{vp_v : v \in D\}$, we have $S \subseteq V'$, and since S clearly satisfies $w'(S) = w(D)$, to complete the proof of the first property it is enough to show that S is stable. With this aim, let vp_v and up_u be distinct vertices in S (which amounts to $v \in D$, $u \in D$ and $v \neq u$). Then $p_v \notin N[u]$ and $p_u \notin N[v]$ because p_v is a private vertex of v in D and p_u is a private vertex of u in D. It follows that vp_v and up_u are not adjacent in G' by the definition of E'. We conclude that S is stable, completing the proof of the first property.

Now we prove the second property, so assume S is stable. Let D be the set $\{u : \text{there exists } v \text{ such that } uv \in S\}$. Note that, since S is stable, if uv and ur belong to S, then we necessarily have $v = r$. It follows that $w(D) = w'(S)$, and so to prove the second property it remains to show that D is irredundant. With this aim in mind, let u be any vertex in D. Then, there exists v such that $uv \in S$. We claim that v is a private vertex of u in D, from which we can conclude that D is irredundant since u is an arbitrary vertex in D. Indeed, since $uv \in S \subseteq V'$, we have $v \in N[u]$. Besides, for any $z \in D - \{u\}$ there exists r such that $zr \in S$, and since S is stable we know that uv and zr are not adjacent in G'. It follows that $v \notin N[z]$ by the definition of E'. This completes the proof of our claim, and in consequence of the second property. □

There is a vast collection of results on the complexity of the MWSS problem; one can navigate the website graphclasses.org, select a graph class and see which complexity class is retrieved from the *Weighted independent set* section, with references included. Some of these graph classes are characterized by forbidden induced subgraphs.

Let us recall that a graph $G_1 = (V_1, E_1)$ is said to be an *induced subgraph* of a graph $G_2 = (V_2, E_2)$, denoted by $G_1 \dot{\subset} G_2$, if there exists an injective map $f : V_1 \to V_2$ that preserves the edge relationship, i.e., such that $(u, v) \in E_1 \Leftrightarrow (f(u), f(v)) \in E_2$. Given a set of graphs $\mathcal{H}$, a family of graphs $\mathcal{G}$ is said to be *$\mathcal{H}$-free* if no graph in $\mathcal{H}$ is an induced subgraph of a graph in $\mathcal{G}$.

Determining the complexity of a problem for $\{G\}$-free graphs, for some (eventually small) G, is the subject of several works. In the case of the MWSS problem, its complexity on $\{G\}$-free graphs (and unrestricted on the weight vector) is known for every graph G with 3 or 4 vertices. Two particular cases where the MWSS problem is polynomial are claw-free [18] and co-paw-free graphs [16], where a *claw* is $K_{1,3}$, and a *co-paw* is the complement of a *paw* (for the latter see Figure 2). We next

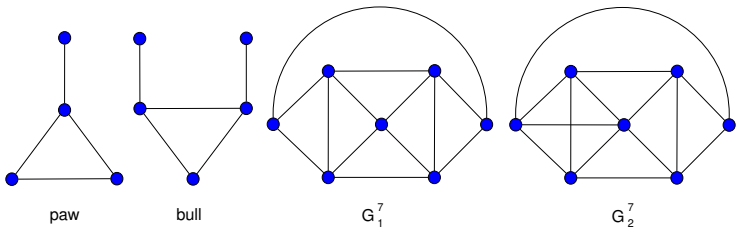

FIGURE 2. A paw, a bull and two graphs of order 7.

use this and the reduction presented above to find graph classes where the MWIS problem is polynomial. This follows from the following results:

Lemma 3.2. *G has a claw, a bull, a P_6, or a $\overline{C_6}$ as induced subgraph if and only if G' has a claw as induced subgraph.*

Lemma 3.3. *G has a co-paw, a G_1^7, or a G_2^7 as induced subgraph if and only if G' has a co-paw as induced subgraph.*

In the statements above, P_6 and C_6 are, respectively, a path and a cycle graph with 6 vertices, and bull, G_1^7 and G_2^7 are the graphs depicted in Figure 2. Therefore, we have:

Theorem 3.4. *Let $X_{\text{claw}} \doteq \{\text{claw}, \text{bull}, P_6, \overline{C_6}\}$ and $X_{\text{copaw}} \doteq \{\text{co-paw}, G_1^7, G_2^7\}$. Then, for any weight vector, the MWIS problem can be solved in polynomial time on X_{claw}-free and X_{copaw}-free graphs.*

Proof. Let G be an X_{claw}-free graph. Then, G' is claw-free by Lemma 3.2. Hence, obtaining $\alpha_w(G')$ is polynomial [18], and so is $\text{IR}_w(G)$ by Theorem 3.1. Using Lemma 3.3 and [16], a similar reasoning can be applied to X_{copaw}-free graphs. □

Remark 3.5. If some of the graph classes considered in Theorem 3.4 were of bounded cliquewidth, then Courcelle's Theorem would imply that the MWIS problem is linear on this class, as mentioned above. To the best of our knowledge, it is not known whether these graph classes are of bounded cliquewidth or not (it is known that some of their superclasses are not, see graphclasses.org). *Even if they were shown to be of bounded cliquewidth, the bound might be high and the algorithm provided by Courcelle's Theorem might be difficult to further analyze.*

Before presenting the proofs of Lemmas 3.2 and 3.3, let us note the following simple results on the induced subgraph relation and the graph transformation introduced above.

Lemma 3.6. *If $H \dot{\subset} G$ then $H' \dot{\subset} G'$.*

Proof. If $H \dot{\subset} G$, there exists an injective map $f : V(H) \to V(G)$ that preserves the edge relationship. Let $f' : V(H') \to V(G')$ be the map defined by $f'(uv) = f(u)f(v)$. Note that the injectivity of f' follows readily from the injectivity of f. So to prove the lemma, it suffices to show that f' preserves the edge relationship.

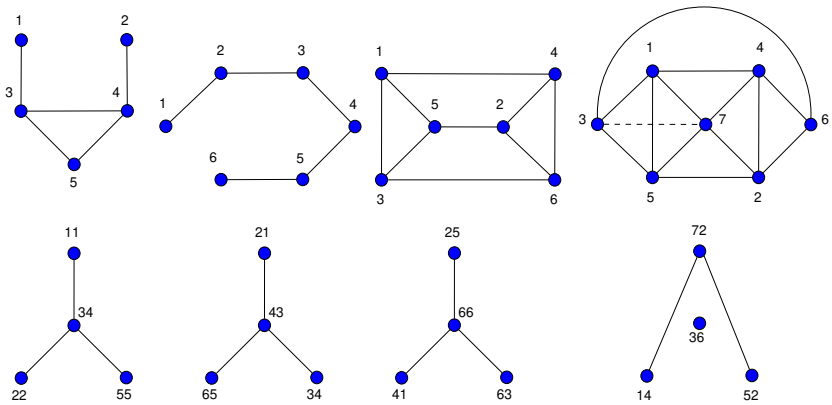

FIGURE 3. A claw in bull′, P_6' and $\overline{C_6}'$, and a co-paw in $G_1^{7\prime}$ and $G_2^{7\prime}$ (note that G_1^7 and G_2^7 only differ in the edge $(3,7)$, which is displayed here by a dashed line).

Indeed, we have $(uv, zr) \in E(H') \Leftrightarrow ((uv \neq zr) \wedge (v \in N[z] \vee r \in N[u])) \Leftrightarrow ((f(u)f(v) \neq f(z)f(r)) \wedge (f(v) \in N[f(z)] \vee f(r) \in N[f(u)])) \Leftrightarrow (f'(uv), f'(zr)) \in E(G')$, where the second equivalence follows from the fact that f is injective and preserves the edge relationship. □

Corollary 3.7. *If $H \dot{\subset} G$ and $K \dot{\subset} H'$, then $K \dot{\subset} G'$.*

Proof. By Lemma 3.6, we have $H' \dot{\subset} G'$, and so $K \dot{\subset} G'$ by the transitivity of the induced subgraph relation. □

Lemma 3.8. *For any graph G, we have $G \dot{\subset} G'$*

Proof. Let $f : V \to V'$ be the map defined by $f(v) = vv$. Since f is clearly injective, to prove the lemma it is enough to show that it preserves the edge relationship. Indeed, we have $(u, v) \in E \Leftrightarrow ((u \neq v) \wedge (u \in N[v])) \Leftrightarrow ((uu \neq vv) \wedge (u \in N[v] \vee v \in N[u])) \Leftrightarrow (uu, vv) \in E'$, which completes the proof. □

We are now ready to present the proofs of Lemmas 3.2 and 3.3.

Proof of Lemma 3.2. We first prove that if G has a claw, a bull, a P_6 or a complement of C_6 as induced subgraph, then claw $\dot{\subset} G'$. In virtue of Corollary 3.7, it is enough to prove that claw $\dot{\subset} H'$ for any $H \in X_{\text{claw}}$ (e.g., if claw $\dot{\subset}$ bull′, then bull $\dot{\subset} G$ implies claw $\dot{\subset} G'$ by this corollary). By Lemma 3.8, we know that claw $\dot{\subset}$ claw′. Besides, Figure 3 displays the vertices of bull′, P_6' and $\overline{C_6}'$, which show that these graphs have a claw as induced subgraph.

To prove the necessity part of Lemma 3.2, suppose that G' has a claw as induced subgraph. Then, since we need to prove that G is not $\{\text{claw}, \text{bull}, P_6, \overline{C_6}\}$-free, the idea is to divide the proof into several cases in each of which we can show that G has a claw, a bull, a P_6 or a complement of C_6 as induced subgraph.

As claw $\dot{\subset} G'$, there exists an injective map $f' : V(\text{claw}) \to V'$ that preserves adjacencies. Without loss of generality, assume that $V(\text{claw}) = \{a, b, c, d\}$ and that a is the *central vertex* of the claw, i.e., that a is adjacent to the other three vertices. If $a_1a_2, b_1b_2, c_1c_2, d_1d_2 \in V'$, with $a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2 \in V$, are the images by f' of the vertices of the claw a, b, c and d, respectively, then we know the following:

(i) For each $x \in \{a, b, c, d\}$, we have $x_1 = x_2$ or x_1 is adjacent to x_2 in G.
(ii) For any $x, y \in \{a, b, c, d\}$ such that $x \neq y$, we have $x_1x_2 \neq y_1y_2$. That is, $x_1 \neq y_1$ or $x_2 \neq y_2$.
(iii) For each $x \in \{b, c, d\}$, we have $a_1 = x_2$ or a_1 is adjacent to x_2 in G or $a_2 = x_1$ or a_2 is adjacent to x_1 in G.
(iv) For any $x, y \in \{b, c, d\}$ such that $x \neq y$, we have $x_1 \neq y_2$, x_1 is not adjacent to y_2 in G, $x_2 \neq y_1$ and x_2 is not adjacent to y_1 in G.

Indeed, (i) follows from the definition of V', (ii) from the injectivity of f', (iii) from the preservation of adjacencies (implying that a_1a_2 is adjacent to b_1b_2, c_1c_2 and d_1d_2 in G'), and (iii) from the preservation of no-adjacencies (implying that b_1b_2, c_1c_2 and d_1d_2 are not adjacent to each other in G'). Conversely, if some vertices $a_1, a_2, b_1, b_2, c_1, c_2, d_1, d_2$ of a graph G satisfy the properties above, then G' has a claw as induced subgraph. We conclude that we need to consider all the possible cases in which the properties above are satisfied, and show that in each of them G has a claw, a bull, a P_6 or a complement of C_6 as induced subgraph.

The simplest case is when $a_1 = a_2$, $b_1 = b_2$, $c_1 = c_2$ and $d_1 = d_2$. In this case, we exhibit a claw in G, i.e., we provide an injective map $f : V(\text{claw}) \to V$ that preserves adjacencies. Let f map the central vertex to a_1 and the others to b_1, c_1 and d_1. By (ii), the vertices a_1, b_1, c_1 and d_1 are different in G, so f is injective. By (iii) and (iv), a_1 is adjacent to b_1, c_1 and d_1 but b_1, c_1 and d_1 are not adjacent to each other, thus f preserves adjacencies.

The next case we consider is when $a_1 = a_2$, $b_1 = b_2$, $c_1 = c_2$ and $d_1 \neq d_2$. Again, we exhibit a claw in G by proposing that f maps the central vertex to a_1, two of the other three vertices to b_1 and c_1, and the remaining one to d_1 if $a_1 = d_2$ or a_1 is adjacent to d_1, or to d_2 if $a_1 = d_1$ or a_1 is adjacent to d_2 (note that one of these conditions must be satisfied by (iii)), so we have four subcases here. We proceed as before and use the properties above to prove that f is injective and preserves adjacencies in each of the resulting cases. Note that, having this proved, by symmetry we could also prove the case $a_1 = a_2$, $b_1 \neq b_2$, $c_1 = c_2$, $d_1 = d_2$, and the case $a_1 = a_2$, $b_1 = b_2$, $c_1 \neq c_2$, $d_1 = d_2$.

The number of cases increases as the equalities $x_1 = x_2$, for $x \in \{a, b, c, d\}$, are replaced by inequalities since this entails the existence of different sets of edges in G and some symmetries may not hold anymore. But this is how a proof can be systematically constructed, see the discussion below. □

Proof of Lemma 3.3. We first prove that if G has a co-paw, a G_1^7 or a G_2^7 as induced subgraph, then co-paw $\dot{\subset} G'$. For this, we use again Corollary 3.7, and conclude that it is enough to prove that co-paw $\dot{\subset} H'$ for any $H \in X_{\text{copaw}}$. Figure 3 points out the vertices of $G_1^{7'}$ and $G_2^{7'}$ which show that these two graphs have a co-paw as induced subgraph. Besides, we know that co-paw $\dot{\subset}$ co-paw$'$ by Lemma 3.8.

As in the case of the proof of Lemma 3.2, to prove the necessity part of Lemma 3.3 we essentially consider all the graphs G such that co-paw $\dot{\subset} G'$, and show that each of them has a co-paw, a G_1^7, or a G_2^7 as induced subgraph. The hypothesis co-paw $\dot{\subset} G'$ is replaced by properties that must be satisfied by a set of vertices of G, similar to the ones given in the proof of the necessity part of Lemma 3.2. Again, the proof is obtained by considering all the possible cases leading to a set of vertices and edges satisfying these properties, see the discussion below. □

As the proofs of the necessity parts of Lemmas 3.2 and 3.3 are based on a large number of cases, it makes little sense to try to present all of them here. Indeed, due to their number, in a traditional paper-and-pencil proof it would be difficult to make sure that all the possible cases are considered and that each one is tackled properly. Moreover, it would be even harder for a reader to check this. Fortunately, nowadays there is a way to increase the level of trust in this kind of results. We think that in this situation the use of a proof assistant turns out to be very convenient (e.g., see [11]).

The *formalization of mathematics* is an area of computer science concerned with expressing and proving mathematical statements in a highly structured language, using a pre-established set of inference rules. Correctness of results is automatically checked by a tool called *proof assistant*. Formalizing proofs provides several benefits, including new ways of visualizing and interacting with the mathematical corpus. Currently, there is a community devoted to formalizing known results of several branches of mathematics. In the case of graph theory, the four color theorem was one of the first long proofs to be formalized in a proof assistant [9], more precisely in Coq.[1] We recall that this theorem was first proved by K. Appel and W. Haken in 1976, and it had the peculiarity that it was necessary to prove thousands of cases, called *configurations*, although this task could be carried out mechanically by a computer. Later, the number of configurations was reduced to 633, but it is still a large number to be considered "manually verifiable". The formalization of such a result in a proof assistant such as Coq has the advantage of reducing trust only to the proof assistant, without involving other software. That is, a "skeptical" reader neither needs to trust in a program that checks all the configurations nor has to write her/his own program to be convinced of the correctness of the theorem, she/he only needs to trust in the proof assistant.

Taking this into account, we proved Lemmas 3.2 and 3.3 with Coq using the extension *Ssreflect* [10] and the formalized graph library developed in [6, 22]. In this regard, it is worth mentioning that this proof assistant let us prove cases that we would have probably missed without it. Besides, it allowed us some kind of automation (which could be probably improved) in the proof of the fact that a set of vertices of a graph induces some subgraph, avoiding us in this way to check repeatedly the same kind of properties, a task that usually leads to mistakes. For this work, we also formalized the first bound of Lemma 2.1, Lemmas 3.6 and 3.8, and Theorem 3.1, among other related results, summarizing 7108 lines of Coq code.

[1]See https://rocq-prover.org/.

This formalization was presented in [15]; we refer the reader to the supplementary material for further details.

4. A heuristic for the MWIS problem

This section is not intended to present a cutting-edge algorithm for the MWIS problem but rather one that works reasonably well and does not need to generate the transformed graph of Theorem 3.1. The experimental results presented below provide evidence that this algorithm would be a good option for instances that are beyond the possibilities of exact approaches, and that it would outperform MWSS heuristics applied to the transformed graph.

We propose a heuristic that delivers upper and lower bounds for $\mathrm{IR}_w(G)$. The upper bound is obtained following Lemma 2.1, computing m_2, while the lower bound is computed by constructing an irredundant set. In order to satisfy the hypothesis of the mentioned lemma, we assume that G does not have any universal vertex. Nevertheless, if G had universal vertices, these could be iteratively removed by virtue of the following observation: if (G, w) satisfies $LB \leq \mathrm{IR}_w(G) \leq UB$, then (G', w') satisfies $\max\{LB, w'(u)\} \leq \mathrm{IR}_{w'}(G') \leq \max\{UB, w'(u)\}$, where G' is the join between $\{u\}$ and G, and $w'(v) = w(v)$ for all vertex v of G; indeed, any irredundant set of G' is either $\{u\}$ or an irredundant set of G.

To present the heuristic, let $G = (V, E)$ and $w \in \mathbb{Z}_+^V$ be the input instance. Start with $\delta_{\min} \leftarrow w(V)$ and $D_{\max} \leftarrow \{u\}$, where u is a vertex with highest weight. For each pair of distinct vertices v_1 and v_2 we compute $s_D(v_1) \leftarrow N[v_1] - N[v_2]$ and $s_D(v_2) \leftarrow N[v_2] - N[v_1]$, and if both of these sets are non-empty we also compute $W \leftarrow N[v_1] \cup N[v_2]$ and its weight, and proceed as follows:

- For each u_1 in $s_D(v_1)$ and u_2 in $s_D(v_2)$, compute $\delta \leftarrow w(W) - w(u_1) - w(u_2)$ and, if $\delta < \delta_{\min}$, update $\delta_{\min} \leftarrow \delta$.
- Assign $D \leftarrow \{v_1, v_2\}$ and $R \leftarrow V - D$. Then, repeat the following until no more vertices can be added to D:
  - Pick $v \in R$ such that $N[v] - W \neq \emptyset$ and $s_D(u) - N[v] \neq \emptyset$ for all $u \in D$. Any v that does not satisfy the previous conditions is removed from R. In case more than one vertex is eligible, pick one that maximizes $\left(1 - \dfrac{|N[v] - W|}{|V|}\right) w(v)$.
  - Update $s_D(v) \leftarrow N[v] - W$, $s_D(u) \leftarrow s_D(u) - N[v]$ for all $u \in D$, $D \leftarrow D \cup \{v\}$, $R \leftarrow R - \{v\}$ and $W \leftarrow W \cup N[v]$.

  Finally, check if $w(D) > w(D_{\max})$ and, in that case, update $D_{\max} \leftarrow D$.

The heuristic returns $m_2 \leftarrow w(V) - \delta_{\min}$ as an upper bound and, since $D_{\max}$ is irredundant by construction, $w(D_{\max})$ as a lower bound of $\mathrm{IR}_w(G)$. In fact, the first item of the procedure above is devoted to compute the minimum in (2.1) since $\delta_{\min}$ coincides with $\delta'_w(G)$ at the end of the execution of the heuristic. On the other hand, the second item repeatedly adds vertices to the set D so that it remains irredundant. There, the procedure keeps track of the vertices dominated by the vertices in D with the set W, the private vertices of the vertices in D with the sets s_D, and the vertices that could be added to D with the set R. Thus, the

condition there allowing the addition of some vertex v to D is equivalent to the fact that $D \cup \{v\}$ is irredundant. When there are several options for the vertex to add, to choose one we apply a criterion aiming at selecting a high-weighted vertex v while, at the same time, the new vertices dominated by v are few, so D has a better chance of being large. The heuristic applies the procedure in the second item starting with $D = \{v_1, v_2\}$ for each pair of vertices v_1 and v_2 such that $s_D(v_1)$ and $s_D(v_2)$ are both non-empty (so $\{v_1, v_2\}$ is irredundant), and keeps the one with highest weight as the output $D_{\max}$.

In order to evaluate the quality of the heuristic, we performed an experiment over some available graphs from the literature used for benchmarking purposes (DIMACS instances up to 100 vertices) taking $w(v) = 1$ for each vertex v, and one instance from a real application (the city of Buenos Aires, in which each vertex models a district and its weight is the population of that district). A computer equipped with an Intel i7-7700 3.6 GHz CPU, 16 GB of RAM, and IBM ILOG CPLEX 12.7 was used. Each run is performed on one thread of the CPU.

For each instance, the heuristic is executed as well as two exact approaches: the first one is CLIQUER [20] that finds a maximum weighted clique of $\overline{G'}$ (and so a maximum weighted stable set of G'), the second one is CPLEX [14] over the integer programming formulation presented below. Thus, both return $\alpha_w(G')$ within an imposed time limit of 1 hour, or fail. In contrast, the heuristic runs in less than half a second for these instances.

In Table 1 we summarize the experiment. Columns 1–3 show the name of the instance, its number of vertices and edge density in percentage. Column 4 is the upper bound m_1 provided by Lemma 2.1. Columns 5–6 are the values delivered by the heuristic. Columns 7–9 display the number of vertices of G', its edge density and weighted independence number. Values in column 6 are highlighted when they are optimal or coincide with the best lower bound available from the exact methods.

As it can be seen in Table 1, the heuristic delivers the optimal solution in almost all instances (the exceptions are 4-Insertions_3 and mug100_25, where $|D_{\max}| = \mathrm{IR}(G) - 1$, and queen9_9, queen8_12 and queen10_10, where no optimal solution is known although the exact algorithms do not provide a better feasible solution than the heuristic). Regarding the upper bounds, m_2 is a little better than m_1, except for queen graphs where the difference is substantial. Both seem to be far from the optimum; however, the additional time consumed by the heuristic dedicated exclusively to obtaining m_2 is negligible with respect to the total time (which is less than half a second).

In order to evaluate the behavior of the heuristic on larger instances (250, 500 and 1000 vertices), we also compared it with a standard MWSS greedy heuristic applied to the transformed graph G' of Theorem 3.1. We generated weighted random instances with different edge densities (the procedure starts with an edgeless graph of $n \in \{250, 500, 1000\}$ vertices and adds edges with a given probability $p \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$; for computing weights, numbers from $\{1, 2, 3, 4, 5\}$ are taken with a uniform distribution). For each instance G, we run the MWIS heuristic

TABLE 1. The experimental results.

Name	$\lvert V(G)\rvert$	dens G	m_1	m_2	$w(D_{\max})$	$\lvert V(G')\rvert$	dens G'	$\alpha_w(G')$
myciel3	11	36.36	8	7	5	51	63.53	5
myciel4	23	28.06	19	18	11	165	54.15	11
queen5_5	25	53.33	13	9	5	345	79.00	5
1-FullIns_3	30	22.99	26	24	14	230	44.15	14
queen6_6	36	46.03	21	15	7	616	71.63	7
2-Insertions_3	37	10.81	34	33	18	181	20.88	18
myciel5	47	21.83	42	41	23	519	47.15	23
queen7_7	49	40.48	31	23	9	1001	65.14	9
2-FullIns_3	52	15.16	48	46	25	454	33.33	25
3-Insertions_3	56	7.14	53	52	27	276	14.09	27
queen8_8	64	36.11	43	33	11	1520	59.64	11
1-Insertions_4	67	10.49	63	62	32	531	22.48	32
huck	74	11.14	73	73	27	676	46.47	27
4-Insertions_3	79	5.06	76	75	38	391	10.13	39
3-FullIns_3	80	10.95	76	74	37	772	27.00	37
jean	80	8.04	80	80	38	588	37.12	38
queen9_9	81	32.59	57	45	13	2193	54.88	–
david	87	10.85	86	86	36	899	61.65	36
mug88_1	88	3.81	85	85	33	380	6.71	33
mug88_25	88	3.81	85	85	33	380	6.66	33
1-FullIns_4	93	13.86	87	85	45	1279	30.87	45
myciel6	95	16.91	89	88	47	1605	41.84	47
queen8_12	96	30.00	71	60	16	2832	51.08	–
mug100_1	100	3.35	97	97	36	432	5.86	36
mug100_25	100	3.35	97	97	36	432	5.85	37
queen10_10	100	29.70	73	59	15	3040	50.79	–
buenosaires	48	10.20	2962	2913	1634	278	18.84	1634

and a greedy heuristic that tries to find a stable set of maximum weight of G'. We also run both heuristics on the real instance.

The greedy heuristic is given as follows.[2] Start with $W \leftarrow V'$ and $S_{\max} \leftarrow \emptyset$. Then, repeat the following until W is empty:

- Pick $uv \in W$ such that $w'(uv)$ is highest. In case of a tie, among those vertices uv having the highest weight, pick one that minimizes $|N(uv) \cap W|$.
- Update $S_{\max} \leftarrow S_{\max} \cup \{uv\}$ and $W \leftarrow W \setminus N[uv]$.

After the loop, $S_{\max}$ is the resulting stable set of G'.

Table 2 reports the results. The best values are highlighted in boldface. A mark "–" means that the algorithm runs out of memory. As one can see from the table, the MWIS heuristic delivers solutions of better quality in all tested instances. The greedy heuristic is faster for low-density graphs; however, for graphs of higher densities, it is unable to allocate the adjacency matrix of G' (in 16 GB of available

[2]In this heuristic, for any $uv \in V'$, $N(uv)$ denotes the neighborhood of uv in G' and $N[uv]$ its closed neighborhood.

TABLE 2. The experimental results on larger instances.

Name	$\lvert V(G)\rvert$	dens G	MWIS heuristic $w(D_{\max})$	MWIS heuristic Time (sec.)	MWSS greedy heuristic $\lvert V(G')\rvert$	MWSS greedy heuristic $w(S_{\max})$	MWSS greedy heuristic Time (sec.)
R250_10	250	10	**184**	13.00	6590	157	**0.28**
R250_30	250	30	**85**	7.18	18988	69	**1.73**
R250_50	250	50	**53**	5.37	31520	40	**4.99**
R250_70	250	70	**35**	**3.63**	43840	25	9.45
R250_90	250	90	**22**	**1.93**	56242	15	19.22
R500_10	500	10	**236**	186	25130	188	**4.5**
R500_30	500	30	**102**	103	75322	82	**61**
R500_50	500	50	**61**	**78**	126016	49	229
R500_70	500	70	**40**	**57**	175202	–	–
R500_90	500	90	**25**	**33**	225006	–	–
R1000_10	1000	10	**284**	2586	101208	243	**136**
R1000_30	1000	30	**113**	**1491**	300518	–	–
R1000_50	1000	50	**69**	**1227**	500294	–	–
R1000_70	1000	70	**45**	**940**	700542	–	–
R1000_90	1000	90	**27**	**575**	899186	–	–
buenosaires	48	10	**1634**	0.01	278	1340	$<$ **0.01**

memory). We expect that, even changing the representation of G' in memory, the greedy heuristic will not scale for larger instances.

If the objective is to solve the MWIS to optimality, an integer programming approach seems to be more promising (as suggested by the results reported in Table 1). An integer programming formulation of the MWSS problem on a graph $G = (V, E)$ is given by

$$\alpha_w(G) = \max\left\{\sum_{v\in V} w(v)x_v : x_u + x_v \leq 1 \,\forall\, (u,v) \in E,\, x_v \in \{0,1\} \,\forall\, v \in V\right\}$$

(note that the solution set of the constraints encodes the stable sets of G). Since any clique of G can contain at most one vertex of a stable set, if $\mathcal{Q}$ is a set of cliques of G that covers every edge of G, then we also have

$$\alpha_w(G) = \max\left\{\sum_{v\in V} w(v)x_v : \sum_{v\in Q} x_v \leq 1 \,\forall\, Q \in \mathcal{Q},\, x_v \in \{0,1\} \,\forall\, v \in V\right\},$$

(because the solution set still encodes the stable sets of G) e.g., see [19]. The advantage of the latter formulation is that it may contain less constraints than the former, and so it might be easier to solve. Applying this to the transformed graph G', we have

$$\alpha_w(G') = \max\left\{\sum_{uv\in V'} w(u)x_{uv} : \sum_{uv\in Q} x_{uv} \leq 1 \,\forall\, Q \in \mathcal{Q},\, x_{uv} \in \{0,1\} \,\forall\, uv \in V'\right\},$$

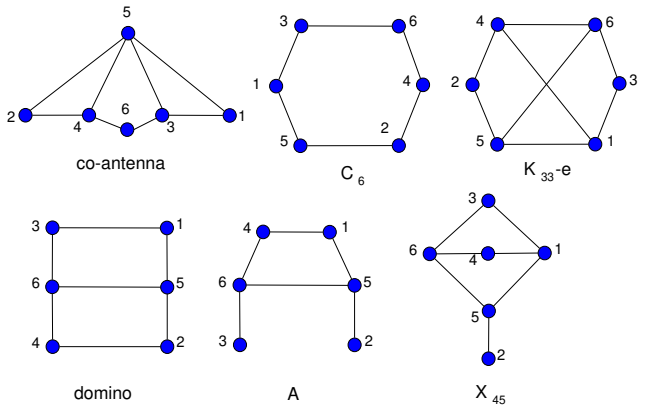

FIGURE 4. A co-antenna, C_6, K_{33}-e, domino, A, and X_{45}.

for any set $\mathcal{Q}$ of cliques of G' that covers every edge of G'. For such set, we propose the next one, which is quadratic in the number of vertices of the original graph G:

$$\mathcal{Q} = \big\{\{uv : v \in N[u] \cap N[z]\} \cup \{zr : r \in N[z]\} : u, z \in V \text{ such that } u \neq z,\ N[u] \cap N[z] \neq \emptyset\big\}.$$

To see that $\mathcal{Q}$ is a set of cliques of G', let u, z be different vertices such that $N[u] \cap N[z] \neq \emptyset$, and define $Q^1_{uz} \doteq \{uv : v \in N[u] \cap N[z]\}$ and $Q^2_z \doteq \{zr : r \in N[z]\}$. Clearly, Q^1_{uz} and Q^2_z are cliques of G'. Moreover, $Q^1_{uz} \cup Q^2_z$ is also a clique of G' since any $uv \in Q^1_{uz}$ is adjacent to any $zr \in Q^2_z$ (due to the fact that $v \in N[z]$). To show that $\mathcal{Q}$ covers every edge of G', let $e = (ab, cd) \in E'$ be an arbitrary edge. Without loss of generality, assume $b \in N[c]$. Note that, since $ab \in V'$, we also have $b \in N[a]$. Then, $Q^1_{ac} \cup Q^2_c$ covers e if $a \neq c$. Otherwise (i.e., if $a = c$), let $u \in N(c)$ (note that $N(c) \neq \emptyset$ because otherwise we would have $a = b = c = d$ contradicting the fact that $e \in E'$, and so that $ab \neq cd$). Then, in this case $Q^1_{uc} \cup Q^2_c$ covers e.

Finally, let us mention that a line for future research may involve the development of cutting-plane algorithms for this formulation, taking advantage of the aforementioned transformation. For instance, since a cycle with five vertices C_5 can contain at most two vertices of a stable set, a family of valid inequalities is $\sum_{uv \in S} x_{uv} \leq 2$ for any $S \subset V'$ that induces a C_5 in G'. However, recognizing structures in G can be cheaper than in G' and, in this case, it can be proven that if G has a C_5, C_6, $K_{3,3}$-e, domino, co-antenna, A, or X_{45} (all these graphs are defined in graphclasses.org and have at most 6 vertices, see Figure 4) as induced subgraph, then G' has a C_5. Indeed, by Corollary 3.7, it is enough to prove that $C_5 \dot{\subset} H'$ for any $H \in \{C_5, C_6, K_{33}\text{-e}, \text{domino}, \text{co-antenna}, A, X_{45}\}$. By Lemma 3.8, we have $C_5 \dot{\subset} C'_5$. If G has a C_6 as induced subgraph with the vertices labeled as in Figure 4, then the vertices 11, 25, 42, 66 and 63 induce a C_5 in G'. We proceed as above with the remaining graphs by exhibiting those vertices that induce a C_5 in G':

- K_{33}-e: 22, 51, 31, 63, 64.
- domino: 11, 51, 22, 64, 63.
- co-antenna: 11, 25, 42, 66, 63.
- A: 11, 14, 63, 36, 52.
- X_{45}: 33, 41, 52, 25, 63.

Acknowledgement

We thank Graciela Nasini for her contributions at the beginning of this work, Mauricio Salichs who helped to transcribe some results in Coq, and Christian Doczkal for his patience with all our questions and requests. We also thank the anonymous reviewers for their comments and suggestions.

References

[1] C. Bazgan, L. Brankovic, K. Casel, and H. Fernau, Domination chain: Characterisation, classical complexity, parameterised complexity and approximability, *Discrete Appl. Math.* **280** (2020), 23–42. DOI MR Zbl

[2] A. Boyaci and J. Monnot, Weighted upper domination number, in *LAGOS'17—IX Latin and American Algorithms, Graphs and Optimization*, Electron. Notes Discrete Math. 62, Elsevier, Amsterdam, 2017, pp. 171–176. DOI MR Zbl

[3] E. J. Cockayne, S. T. Hedetniemi, and D. J. Miller, Properties of hereditary hypergraphs and middle graphs, *Canad. Math. Bull.* **21** no. 4 (1978), 461–468. DOI MR Zbl

[4] B. Courcelle, J. A. Makowsky, and U. Rotics, Linear time solvable optimization problems on graphs of bounded clique-width, *Theory Comput. Syst.* **33** no. 2 (2000), 125–150. DOI MR Zbl

[5] M. Cygan, F. V. Fomin, Ł. Kowalik, D. Lokshtanov, D. Marx, M. Pilipczuk, M. Pilipczuk, and S. Saurabh, *Parameterized algorithms*, Springer, Cham, 2015. MR Zbl

[6] C. Doczkal and D. Pous, Graph theory in Coq: minors, treewidth, and isomorphisms, *J. Automat. Reason.* **64** no. 5 (2020), 795–825. DOI MR Zbl

[7] R. G. Downey, M. R. Fellows, and V. Raman, The complexity of irredundant sets parameterized by size, *Discrete Appl. Math.* **100** no. 3 (2000), 155–167. DOI MR Zbl

[8] O. Favaron, T. W. Haynes, S. T. Hedetniemi, M. A. Henning, and D. J. Knisley, Total irredundance in graphs, *Discrete Math.* **256** no. 1-2 (2002), 115–127. DOI MR Zbl

[9] G. Gonthier, Formal proof—the four-color theorem, *Notices Amer. Math. Soc.* **55** no. 11 (2008), 1382–1393. MR Zbl

[10] G. Gonthier and A. Mahboubi, An introduction to small scale reflection in Coq, *J. Formaliz. Reason.* **3** no. 2 (2010), 95–152. DOI MR Zbl

[11] J. Harrison, Formal proof—theory and practice, *Notices Amer. Math. Soc.* **55** no. 11 (2008), 1395–1406. MR Zbl

[12] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater, *Fundamentals of domination in graphs*, Monogr. Textbooks Pure Appl. Math. 208, Marcel Dekker, New York, 1998. MR Zbl

[13] T. W. Haynes, S. T. Hedetniemi, and P. J. Slater (eds.), *Domination in graphs: Advanced topics*, Monogr. Textbooks Pure Appl. Math. 209, Marcel Dekker, New York, 1998. MR Zbl

[14] IBM ILOG CPLEX Optimizer. Available at https://www.ibm.com/products/ilog-cplex-optimization-studio/cplex-optimizer.

[15] R. D. Katz and D. Severín, Coq/Ssreflect for large case-based graph theory proofs, 2021, The 12th Coq Workshop, online, affiliated with ITP 2021. Abstract available at https://coq-workshop.gitlab.io/2021/abstracts/Coq2021-04-02-graph-theory-proofs.pdf

[16] V. V. Lozin and R. Mosca, Independent sets in extensions of $2K_2$-free graphs, *Discrete Appl. Math.* **146** no. 1 (2005), 74–80. DOI MR Zbl

[17] A. A. McRae, *Generalizing NP-completeness proofs for bipartite graphs and chordal graphs*, Ph.D. thesis, Clemson University, 1994. Available at https://open.clemson.edu/arv_dissertations/1682.

[18] G. J. Minty, On maximal independent sets of vertices in claw-free graphs, *J. Combin. Theory Ser. B* **28** no. 3 (1980), 284–304. DOI MR Zbl

[19] G. L. Nemhauser and G. Sigismondi, A strong cutting plane/branch-and-bound algorithm for node packing, *J. Oper. Res. Soc.* **43** (1992), 443–457. DOI Zbl

[20] P. R. J. Östergård, A new algorithm for the maximum-weight clique problem, *Nordic J. Comput.* **8** no. 4 (2001), 424–436. MR Zbl

[21] P. Pinacho Davidson, C. Blum, and J. A. Lozano, The weighted independent domination problem: Integer linear programming models and metaheuristic approaches, *European J. Oper. Res.* **265** no. 3 (2018), 860–871. DOI MR Zbl

[22] D. E. Severín, Formalization of the domination chain with weighted parameters, in *10th International Conference on Interactive Theorem Proving*, LIPIcs. Leibniz Int. Proc. Inform. 141, Schloss Dagstuhl. Leibniz-Zentrum für Informatik, Wadern, 2019, article no. 36. DOI MR Zbl

Ricardo D. Katz
CIFASIS-CONICET, Argentina
katz@cifasis-conicet.gov.ar

Daniel Severín✉
Departamento de Matemática, FCEIA, Universidad Nacional de Rosario, Argentina
CONICET, Argentina
daniel@fceia.unr.edu.ar

Received: May 18, 2022
Accepted: July 18, 2023
Early view: August 24, 2024