Electronic Colloquium on Computational Complexity, Report No. 143 (2026)

The Complexity of Boolean-Weighted Graph Isomorphism

Francesco Cristiano*

Abstract. A Boolean-weighted graph is a finite graph whose edges carry DNF formulas over a common variable set. This paper studies two isomorphism problems on such graphs, distinguished by whether the per-edge condition requires the matched edge labels to be syntactically DNF-isomorphic (BWG-ISO) or syntactically DNF non-isomorphic (BWG-NI), each over a vertex bijection that preserves edges. The two sit at opposite ends of one phenomenon: imposing local syntactic DNF-isomorphism on every edge does not raise complexity above graph isomorphism, whereas replacing it by local non-isomorphism makes the problem NP-hard, the jump being driven by the search over host isomorphisms rather than by the complexity of the labels. For the isomorphism requirement, BWG-ISO $\equiv_m^p$ GI: although syntactic DNF isomorphism is itself GI-complete [ACL12], attaching such a label to every edge leaves the global problem exactly at GI. For the non-isomorphism requirement, the per-edge test is a graph-non-isomorphism query, placing BWG-NI in $\mathrm{NP}^{\mathrm{GI}}$; the paper shows it is GI-hard and GNI-hard and not coNP-hard unless $\mathrm{PH} = \Sigma_2^{\mathrm{P}}$, and proves it is NP-hard. The hardness already holds when labels are restricted to monotone DNFs, equivalently $\{0,1\}$-matrices, so it stems from the interaction with host automorphisms, not from rich Boolean structure. This NP-hardness is unconditional; the hardest NP set known to $\leq_m^p$-reduce to FI is GI [AT00]. BWG-NI is thereby a natural inhabitant of $\mathrm{NP}^{\mathrm{GI}}$: with the NP-hardness, $\mathrm{NP}^{\mathrm{GI}}$ membership places it outside P^{GI}, $\mathrm{coNP}^{\mathrm{GI}}$, and Low_2 unless $\mathrm{PH} = \Sigma_2^{\mathrm{P}}$, making it a natural candidate to separate $\mathrm{NP}^{\mathrm{GI}}$ from P^{GI} when the hierarchy is infinite. The second-level collapse that would be forced by its Σ_2^{P}-completeness is one level below the third-level collapse Agrawal and Thierauf obtain for the formula isomorphism problem FI, reflecting that BWG-NI, unlike the coNP-hard FI, lies in $\mathrm{NP}^{\mathrm{GI}}$. The counting problem #BWG-NI is #P-hard.

1. Introduction

Two isomorphism problems delimit the structural complexity of equivalence under variable renaming. Graph isomorphism GI lies in NP $\cap$ coAM, is low for Σ_2^{P} [Sch88], and is not NP-complete unless the polynomial hierarchy collapses. Formula isomorphism FI [AT00] lies in Σ_2^{P}, is coNP-hard, and is not Σ_2^{P}-complete unless the hierarchy collapses. The two are connected by the factorisation FI $= \exists\lambda\cdot$EQ of [Cri26], an existential over renamings applied to a Boolean-equivalence test, and by the result of [ACL12] that syntactic DNF isomorphism (DSFI) is GI-complete.

A Boolean-weighted graph (BWG) consists of a host graph whose edges carry DNF formulas over a fixed variable set. Fixing a host bijection, one may impose on each matched pair of edge labels either of two conditions: that they be DSFI-isomorphic or that they be DSFI non-isomorphic. The two conditions yield two problems, the isomorphism variant BWG-ISO and the non-isomorphism variant BWG-NI, of markedly different complexity. Attention is restricted throughout to general (unrestricted) host graphs.

For the isomorphism variant, Section 3 proves BWG-ISO $\equiv_m^p$ GI: the per-edge isomorphism requirements can be encoded as a single coloured-graph isomorphism query, each labelled edge being replaced by a canonical anchor vertex carrying the label's gadget. The isomorphism variant therefore sits exactly at the GI degree, hence is low for Σ_2^{P}; the per-edge labels, each of which is GI-complete in isolation [ACL12], are absorbed into a single GI instance without raising the complexity.

For the non-isomorphism variant, the picture is different. The per-edge condition is a graph non-isomorphism (GNI) test, which places BWG-NI in $\mathrm{NP}^{\mathrm{GI}}$ (Proposition 4.1). Section 4 also shows BWG-NI is GI-hard and GNI-hard but not coNP-hard unless $\mathrm{PH} = \Sigma_2^{\mathrm{P}}$, and Section 5 proves it is NP-hard by a reduction from the fixed-point-free automorphism problem FPFAUT; the NP-hardness is carried by the host-automorphism

*fra.cristiano@gmail.com

ISSN 1433-8092

search, not by the labels. $\mathrm{NP}^{\mathrm{GI}}$ membership together with the NP-hardness then gives (Sections 6–7) that BWG-NI is not Σ_2^{P}-complete and not $\leq_m^p$-equivalent to FI unless the hierarchy collapses, and that it lies in none of P^{GI}, $\mathrm{coNP}^{\mathrm{GI}}$, Low_2 unless the hierarchy collapses, while the rigid-host restriction (BWG-NI on hosts with no nontrivial automorphism) is low for Σ_2^{P}, and GNI-complete (equivalently, $\leq_m^p$-complete for the conjunctive closure of GNI) on rigid hosts drawn from the tractable classes of Section 7. The counting problem #BWG-NI is shown to be #P-hard.

Read together, the two variants localize the complexity: the existential search over host isomorphisms (trivial precisely when the host is rigid) is what lifts it to NP-hardness while the GI-complete labels keep BWG-NI inside $\mathrm{NP}^{\mathrm{GI}}$, so BWG-NI is a natural candidate inhabitant of $\mathrm{NP}^{\mathrm{GI}}$ lying outside P^{GI} unless the hierarchy collapses. This membership also sharpens the comparison with FI: were BWG-NI Σ_2^{P}-complete the hierarchy would collapse to its second level, one step below the third-level collapse entailed by Σ_2^{P}-completeness of FI [AT00], so BWG-NI obstructs second-level completeness more strongly than the coNP-hard FI, a second-level collapse being the stronger consequence since it implies the third-level one but not conversely.

All conditional separations obtained here are of relativizing type and do not yield unconditional separations.

Contributions. The following table summarises the results; the third column gives the section in which each is established.

Result	Statement	Section
Isomorphism variant	BWG-ISO $\equiv_m^p$ GI	3
Membership	BWG-NI $\in \mathrm{NP}^{\mathrm{GI}}$	4
Hardness profile	GI-hard and GNI-hard; not coNP-hard unless $\mathrm{PH} = \Sigma_2^{\mathrm{P}}$	4
NP-hardness	NP-hard via FPFAUT	5
NP-hardness vs FI	BWG-NI NP-hard unconditionally; FI not known NP-hard, its known deterministic floor from NP only GI [AT00]	5, 6
Counting	#BWG-NI is #P-hard, in $\#\cdot\mathrm{P}^{\mathrm{GI}}$	5
Non-completeness	not Σ_2^{P}-complete and not $\equiv_m^p$ FI unless $\mathrm{PH} = \Sigma_2^{\mathrm{P}}$	6
General hosts	NP-hard $\Rightarrow \notin \mathrm{P}^{\mathrm{GI}}/\mathrm{coNP}^{\mathrm{GI}}/\mathrm{Low}_2$ unless collapse; P^{GI}-membership $\Rightarrow \mathrm{NP} \subseteq \mathrm{SPP}$	7
Rigid hosts	rigid-host BWG-NI is low for Σ_2^{P}, GNI-complete on tractable rigid hosts (Theorem 7.2)	7
Label alphabet	monotone, two-symbol labels suffice for all of the above	8

The per-edge label-comparison relation is the syntactic formula-isomorphism relation of [ACL12], a GI-complete equivalence relation, so BWG-NI is a two-level isomorphism problem: an isomorphism test on structures whose components are themselves compared under isomorphism; comparing labels by plain string disequality instead yields unconditional NP-completeness (Remark 8.2). The host quantifier alone positions the problem across its range: with the per-edge comparison held fixed, the rigid-host restriction is GNI-complete (Theorem 7.2), while admitting hosts with nontrivial automorphisms raises the problem from the GNI degree to NP-hardness (Proposition 7.5).

Organisation. Section 2 fixes notation and records the anchoring facts. Section 3 establishes the isomorphism variant, BWG-ISO $\equiv_m^p$ GI. Sections 4–7 develop the non-isomorphism variant: membership and hardness profile (4), NP-hardness and the consequent counting hardness (5), non-completeness and non-equivalence with FI (6), and the degree, lowness, and host-automorphism analysis (7). Section 8 records the label-alphabet refinement and collects observations and a map of the resulting complexity landscape. Section 9 lists caveats and open problems.

2. Preliminaries

Familiarity with the polynomial hierarchy is assumed. GI and GNI denote graph isomorphism and graph non-isomorphism; GI $\in$ NP, GNI $\in$ coNP $\cap$ AM [GMW91], and GI $\in$ Low_2, i.e. $\Sigma_2^{\mathrm{GI}} = \Sigma_2^{\mathrm{P}}$ [Sch88], where $\mathrm{Low}_2 = \{ A : \Sigma_2^A = \Sigma_2^{\mathrm{P}} \}$ consists of the sets that give Σ_2^{P} no additional power when used as oracles. FI is the formula isomorphism problem; FI $\in \Sigma_2^{\mathrm{P}}$ and is coNP-hard [AT00]. EQ is Boolean equivalence, which is coNP-complete. DSFI is syntactic DNF isomorphism, which is GI-complete [ACL12]; its negation DSNI $= \neg$DSFI is GNI-complete. $\mathrm{NP}^{\mathrm{GI}}$ (resp. P^{GI}) is the class decided by an NP (resp. deterministic polynomial-time) machine with a GI oracle; both are closed under $\leq_m^p$, and $\mathrm{NP}^{\mathrm{GI}} \subseteq \mathrm{NP}^{\mathrm{NP}} = \Sigma_2^{\mathrm{P}}$. For a graph G, Aut(G) denotes its automorphism group, and G is rigid (asymmetric) if Aut(G) is trivial, that is, if its only automorphism is the identity.

Fix a variable set Z. A Boolean-weighted graph (BWG) is a pair A $= (H_A, \ell_A)$ in which $H_A = (V_A, E_A)$ is a finite simple graph, the host graph of A, and $\ell_A : E_A \to$ DNF(Z) is the edge labelling, assigning to each edge a DNF formula over Z. The host graph carries the combinatorial structure on which candidate isomorphisms act; the label of an edge is the additional per-edge datum that such an isomorphism must respect. The relation imposed on a matched pair of labels (DSFI in BWG-ISO, DSNI in BWG-NI) is the per-edge comparison. The unrestricted (general) case places no restriction on the host; a restricted host family (for instance, the rigid hosts of Section 7) constrains which graphs may serve as the host. For an edge e $= \{$u, v$\}$ and a bijection h, write h(e) $= \{$h(u), h(v)$\}$; write n $= |V_A|$, m $= |E_A|$, and $\langle$A$\rangle$ for the input size; for a BWG G we also write V(G) and E(G) for the vertex and edge sets of its host. Two DNF formulas φ, ψ are syntactically DNF-isomorphic when one is carried to the other by a renaming of the variables together with a reordering of the terms and of the literals within each term; DSFI denotes the associated decision problem, the set of pairs (φ, ψ) that are so related, so (φ, ψ) $\in$ DSFI records that φ and ψ are syntactically DNF-isomorphic.

In the two definitions that follow, the clauses are tagged (S) for the structural condition on the hosts and (L^+), (L^-) for the label condition in its positive and negative form; the two problems share (S) and differ only in the sign of the label clause.

Definition 2.1 (BWG-ISO). Given BWGs A, B with $|V_A| = |V_B|$, the pair (A, B) lies in BWG-ISO iff there is a bijection h : $V_A \to V_B$ that is (S) a graph isomorphism $H_A \to H_B$ and (L^+) satisfies (ℓ_A(e), ℓ_B(h(e))) $\in$ DSFI for every e $\in E_A$; equivalently $\exists$h [h is a graph isomorphism between the host graphs H_A and H_B $\wedge\forall$e : (ℓ_A(e), ℓ_B(h(e))) $\in$ DSFI].

Definition 2.2 (BWG-NI). Given BWGs G, H with $|V_G| = |V_H|$, the pair (G, H) lies in BWG-NI iff there is a bijection f : $V_G \to V_H$ that is (S) a graph isomorphism $H_G \to H_H$ and (L^-) satisfies (ℓ_G(e), ℓ_H(f(e))) $\notin$ DSFI for every e $\in E_G$; equivalently $\exists$f [f is a graph isomorphism between the host graphs H_G and H_H $\wedge\forall$e : (ℓ_G(e), ℓ_H(f(e))) $\notin$ DSFI].

Three facts anchor the development, stated as Facts 2.3–2.5 and referred to by number throughout.

Fact 2.3 (the syntactic comparison is GI-complete [ACL12]). DSFI is GI-complete. Informally, gad turns each DNF formula into a plain finite graph, so that two formulas yield isomorphic graphs exactly when they are syntactically DNF-isomorphic. More formally, there is a polynomial-time map gad : DNF(Z) $\to \mathscr{G}$, where $\mathscr{G}$ is the class of all finite graphs, such that gad(φ) has size polynomial in the length of φ, and, for all φ, $\psi \in$ DNF(Z), gad(φ) $\cong$ gad(ψ) $\Leftrightarrow$ (φ, ψ) $\in$ DSFI. Concretely, gad is the composition of two reductions of [ACL12]. First, φ is written as its variable-by-term incidence matrix over $\{-1,0,1,2\}$, recording for each variable and term whether the variable is absent, occurs unnegated, occurs negated, or occurs both ways. Second, that matrix is turned into a graph with a vertex for each variable, each term, and each matrix entry, together with fixed cliques of distinct sizes that mark the variable side and the term side and that codify the four entry values; each entry-vertex is joined to its variable, to its term, and by a single edge to a designated vertex of the clique codifying its value (Figure 1). By [ACL12, Theorem 2] (Lemmas 4 and 5 there) two such graphs are isomorphic iff the incidence matrices agree up to row and column permutations, that is, iff the formulas are related by a variable renaming together with a term reordering: syntactic DNF isomorphism. The monotone fragment (formulas with no negated variables) is already GI-complete [ACL12]. Hence DSNI is GNI-complete and DSFI $\equiv_m^p$ GI.

Throughout, gad denotes the map of Fact 2.3, sending each DNF formula to the plain graph gad(φ) [ACL12]; it supplies the per-label graphs compared in both variants. The construction that turns it into a global reduction object is developed in Section 3 and is referred to as the coloured incidence encoding.

Fact 2.4 (the semantic comparison [AT00]). FI is coNP-hard and lies in Σ_2^{P}; it is the semantic analogue of DSFI, asking for a renaming making two formulas logically equivalent. Its negation FNI = co-FI is NP-hard and lies in Π_2^{P}. Moreover GI $\leq_m^p$ FI [BRS98, Proposition 12], a reduction observed with R. Chang [BR93, Proposition 2] whose image $\bigvee_{(i,j)\in E}(v_i \wedge v_j)$ is itself a monotone DNF, so it lands in FI directly; independently, semantic and syntactic isomorphism coincide on canonical DNF inputs (minterm expansions), where GI $\equiv_m^p$ CFI, canonical formula isomorphism, the restriction of FI to such inputs, and the identity map on this restriction gives GI $\leq_m^p$ CFI $\leq_m^p$ FI [Cri26, Theorem 2].

Fact 2.5 (GI is low in the polynomial hierarchy [Sch88]). GI $\in$ Low$_2$, hence $\mathrm{P}^{\mathrm{GI}} \subseteq$ Low$_2 \subseteq \Sigma_2^{\mathrm{P}} \cap \Pi_2^{\mathrm{P}}$. By contrast $\mathrm{P}^{\mathrm{FI}} \supseteq$ coNP (by Fact 2.4), so $\mathrm{P}^{\mathrm{GI}} \subseteq \mathrm{P}^{\mathrm{FI}}$, and the inclusion is strict unless PH collapses, since coNP $\subseteq \mathrm{P}^{\mathrm{GI}}$ would give coNP $\subseteq$ Low$_2$, i.e. PH $= \Sigma_2^{\mathrm{P}}$.

Lemma 2.6 (label precomputation). Let the edges of an instance carry labels from a set S with |S| polynomial in the input. With O($|S|^2$) GI queries one can compute an index ind : S $\to \mathbb{N}$ such that two labels receive the same index iff they are syntactically DNF-isomorphic. After this precomputation every per-edge comparison is the integer test ind(ℓ(e)) $\neq$ ind(ℓ(e')) (for the isomorphism variant, ind(ℓ(e)) = ind(ℓ(e'))), which uses no oracle.

Proof. There are at most |S| distinct labels; compare every pair (φ, ψ) with one GI query on (gad(φ), gad(ψ)), correct by Fact 2.3, and take the equivalence classes of the resulting relation as index classes. The number of pairs is O($|S|^2$), and the resulting integers determine every per-edge comparison by equality with no further oracle access. ∎

Definition 2.7 (precomputable-query fragment $\mathrm{NP}^{\mathrm{GI}}_{\mathrm{pre}}$). An $\mathrm{NP}^{\mathrm{GI}}$ computation has precomputable queries if every instance submitted to the oracle lies in a polynomial-size set determined by the input alone, independently of the nondeterministic guess; the oracle answers can then be tabulated in advance (Lemma 2.6) and the nondeterministic phase made oracle-free. Let $\mathrm{NP}^{\mathrm{GI}}_{\mathrm{pre}}$ be the class decided by such a computation. BWG-NI $\in \mathrm{NP}^{\mathrm{GI}}_{\mathrm{pre}}$: the only label graphs that occur are gad(ℓ(e)) over the edges of the two inputs, so every comparison lies among the O(m^2) input-fixed pairs, and a guessed f selects which of these are required to be non-isomorphic. The inclusion $\mathrm{NP}^{\mathrm{GI}}_{\mathrm{pre}} \subseteq \mathrm{NP}^{\mathrm{GI}}$ is immediate; whether it is proper is open.

CGNI$^\wedge$ denotes the closure of GNI under polynomial conjunction $\bigwedge_i$ GNI(A_i, B_i); the class is defined precisely in Definition 7.1. Since GNI is AND-closed (Definition 7.1) it coincides with the $\leq_m^p$-degree of GNI, and in particular lies in the Boolean hierarchy over GI.

3. The isomorphism variant: BWG-ISO is equivalent to GI

This section treats the isomorphism variant and reduces it to a single graph-isomorphism query. The obstruction to a per-edge reduction is that gad carries no isomorphism-invariant distinguished vertex (Fact 2.3): rooting at an arbitrary vertex of a label graph is not isomorphism-invariant. The remedy is a canonical per-edge anchor, external to the label graphs, onto which each label graph is coned.

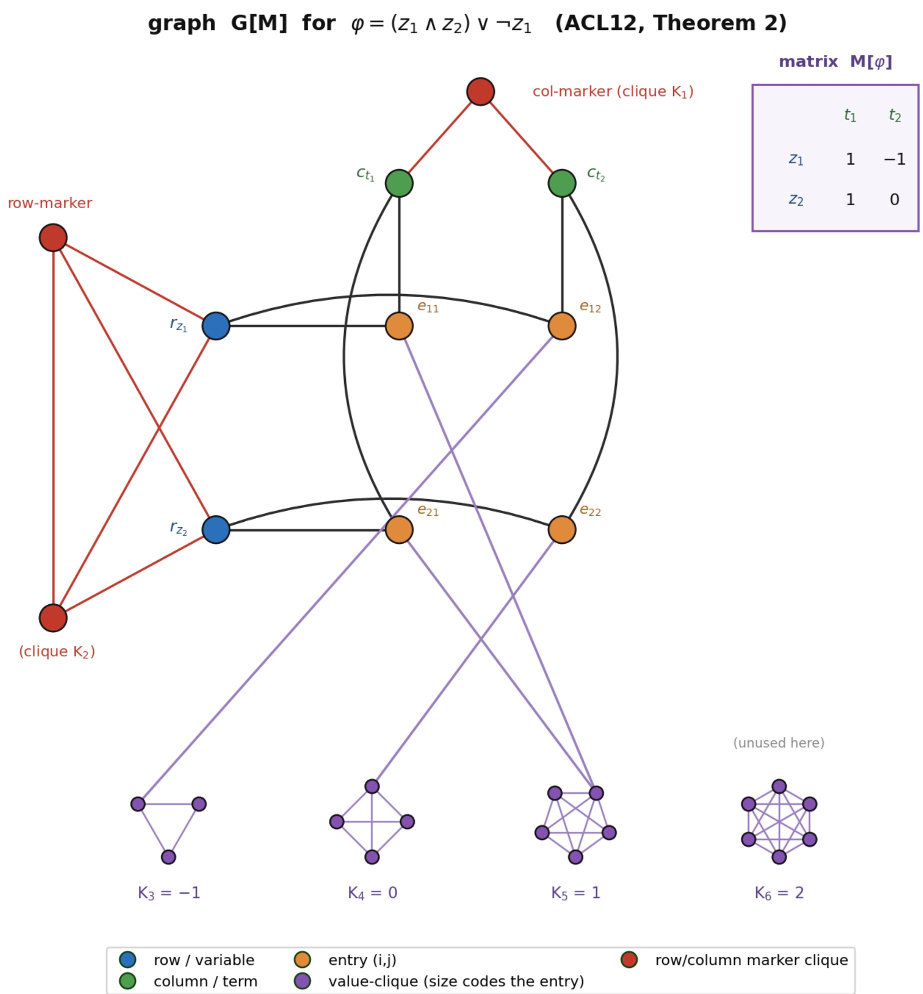

Figure 1. *The gadget gad(φ) of Fact 2.3 for the non-monotone formula $\varphi = (z_1 \wedge z_2) \vee \neg z_1$, following [ACL12]. The formula is written as its variable-by-term matrix over $\{-1, 0, 1, 2\}$ (top right); the graph has one vertex per variable, per term, and per matrix entry, with marker cliques of sizes 1 and 2 separating terms from variables and value-cliques of sizes 3, 4, 5, 6 codifying the entries $-1, 0, 1, 2$. Each entry-vertex joins its variable, its term, and, by a single edge, a designated vertex of the clique of its value; by [ACL12, Theorem 2] graph isomorphism of these gadgets is exactly a variable renaming with a term reordering.*

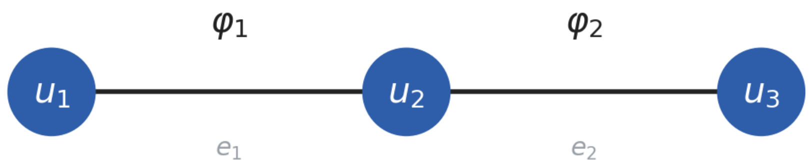

Figure 2. *A Boolean-weighted graph A: host path u_1–u_2–u_3 with edge labels $\varphi_1 = (z_1 \wedge z_2) \vee (z_2 \wedge z_3)$ and $\varphi_2 = z_1 \wedge z_3$.*

Construction 3.1 (coloured incidence encoding). The DNF label on each host edge e is encoded by a gadget Γ_e, the graph that the map gad of Fact 2.3 builds from that formula. More formally, for a BWG A (Figure 2) and each edge e $\in E_A$, write $\Gamma_e :=$ gad(ℓ_A(e)). Define a vertex-coloured graph Φ(A) = (W_A, F_A, κ) over the colour set {h, m, g}. The vertex set W_A consists of the host vertices, one new edge-vertex m_e per host edge e, and the vertices of the gadget copies Γ_e, all pairwise disjoint; more formally, $W_A = V_A \uplus \{m_e : e \in E_A\} \uplus \bigsqcup_e V(\Gamma_e)$, where $\uplus$ and $\bigsqcup$ denote disjoint union and V(Γ_e) is the vertex set of Γ_e. The colouring is κ = h on V_A, κ = m on the edge-vertices m_e, and κ = g on each gadget copy (gad returns a plain graph, so a single gadget colour suffices). The edge set F_A has three types: (incidence) u $\sim m_e$ iff u $\in$ e, so V_A together with the edge-vertices m_e forms the incidence graph of H_A; (cone) $m_e \sim$ a for every a $\in$ V(Γ_e), i.e. each gadget vertex is joined to the apex m_e (the cone over Γ_e with apex m_e); (gadget) the edges of each Γ_e. Write Φ(A) $\cong_{\text{col}}$ Φ(B) when a colour-preserving isomorphism Φ(A) $\to$ Φ(B) exists.

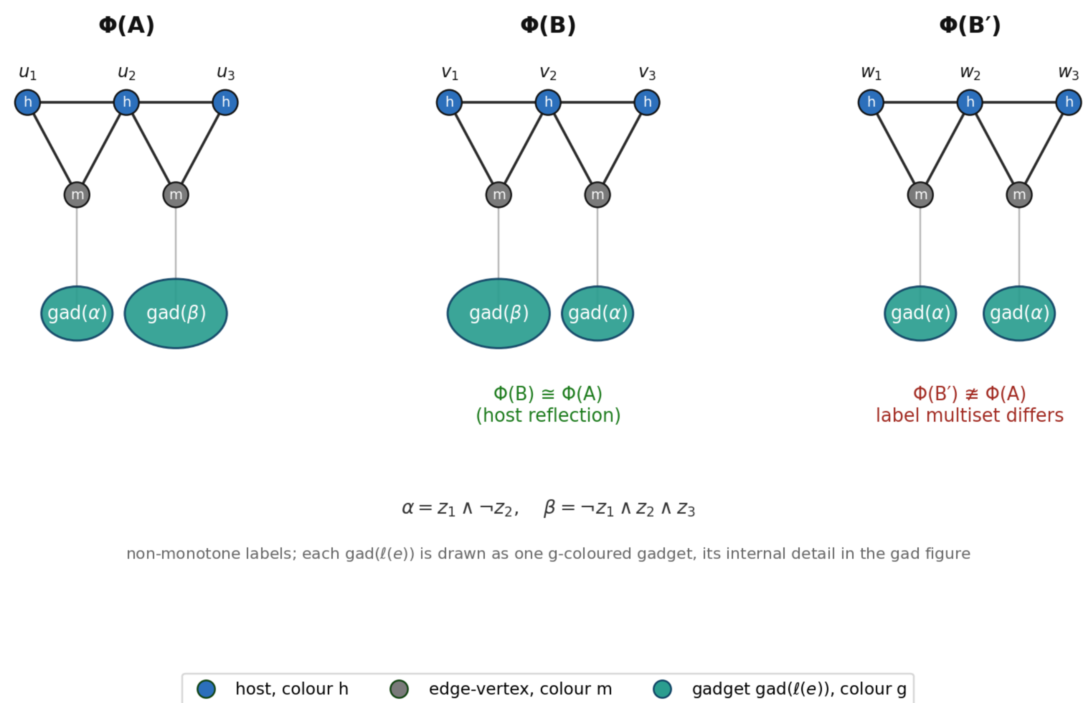

Figure 3. *The reduction of Construction 3.1 on a small instance, with each edge's gadget drawn as a single g-coloured block gad($\ell(e)$) whose internal structure is that of Figure 1. The pair (A, B) is isomorphic by a host reflection, so Φ(A) $\cong_{\text{col}}$ Φ(B); the pair (A, B') differs in its label multiset, so Φ(A) $\ncong$ Φ(B'). Labels $\alpha = z_1 \wedge \neg z_2$ and $\beta = \neg z_1 \wedge z_2 \wedge z_3$; the colour classes h, m, g separate host, incidence, and gadget.*

Definition 3.2 (edge-anchor gadget). For e $\in E_A$, let N_A(e) be Φ(A) with the single vertex m_e individualised, that is, assigned a fresh colour given to no other vertex. The anchor m_e is canonical: it is the unique m-vertex whose h-neighbourhood equals e, since H_A is simple and distinct edges have distinct endpoint sets.

Lemma 3.3 (gadget isolation and isomorphism transfer). In Φ(A): (i) the g-neighbourhood of m_e is exactly V(Γ_e), and gadget vertices are adjacent only within their own gadget and to their apex m_e; (ii) for any colour-preserving isomorphism $\varphi : \Phi(\mathrm{A}) \to \Phi(\mathrm{B})$ with $\varphi(m_e) = m_{e'}$, the restriction $\varphi\restriction$V(Γ_e) is a colour-preserving isomorphism $\Gamma_e \to \Gamma_{e'}$.

Proof. (i) is immediate from Construction 3.1: a vertex of Γ_e has neighbours only inside V(Γ_e) (gadget edges) and $\{m_e\}$ (cone edges); distinct gadgets are mutually non-adjacent and touch no h-vertex. (ii) Since $\varphi(m_e)$ $= m_{e'}$, φ carries the apex's g-neighbourhood V(Γ_e) bijectively onto V($\Gamma_{e'}$); by (i) the only edges among these vertices are gadget-internal and are preserved by φ, so the restriction is an isomorphism $\Gamma_e \to \Gamma_{e'}$. ■

Construction 3.4 (decolouring [KST93]). Let $\hat{\Phi}$(A) be obtained from Φ(A) by attaching to every vertex a tag that encodes its colour, using the individualisation gadget of [KST93]: with the three colours indexed i $\in$ {1, 2, 3} and N = $\max(|V(\Phi(\mathrm{A}))|, |V(\Phi(\mathrm{B}))|)$ a common value for the pair (if this maximum is below 4 the hosts are edgeless and the instance is decided directly), a colour-i vertex v receives a fresh path of $2N+3$ vertices joined at one end to v, together with a secondary path of i further vertices branching from the $(N+2)$-th vertex of that chain. These tags are pairwise non-isomorphic across colours and identifiable by their chain length; each tag, rooted at its attachment vertex v, admits no nontrivial automorphism fixing v, since, measured from the branch vertex, the two arms of the chain have different lengths (N + 2 through v against N + 1 to the far leaf), breaking the reflection of the bare chain even when v is isolated, and none occurs elsewhere in $\hat{\Phi}$(A), so no isomorphism of $\hat{\Phi}$(A) can carry a tag vertex off its tag or across colours; hence $\hat{\Phi}(\mathrm{A}) \cong \hat{\Phi}(\mathrm{B}) \Leftrightarrow \Phi(\mathrm{A}) \cong_{\mathrm{col}} \Phi(\mathrm{B})$.

Figure 4 shows this discharge on one edge-vertex and its gadget.

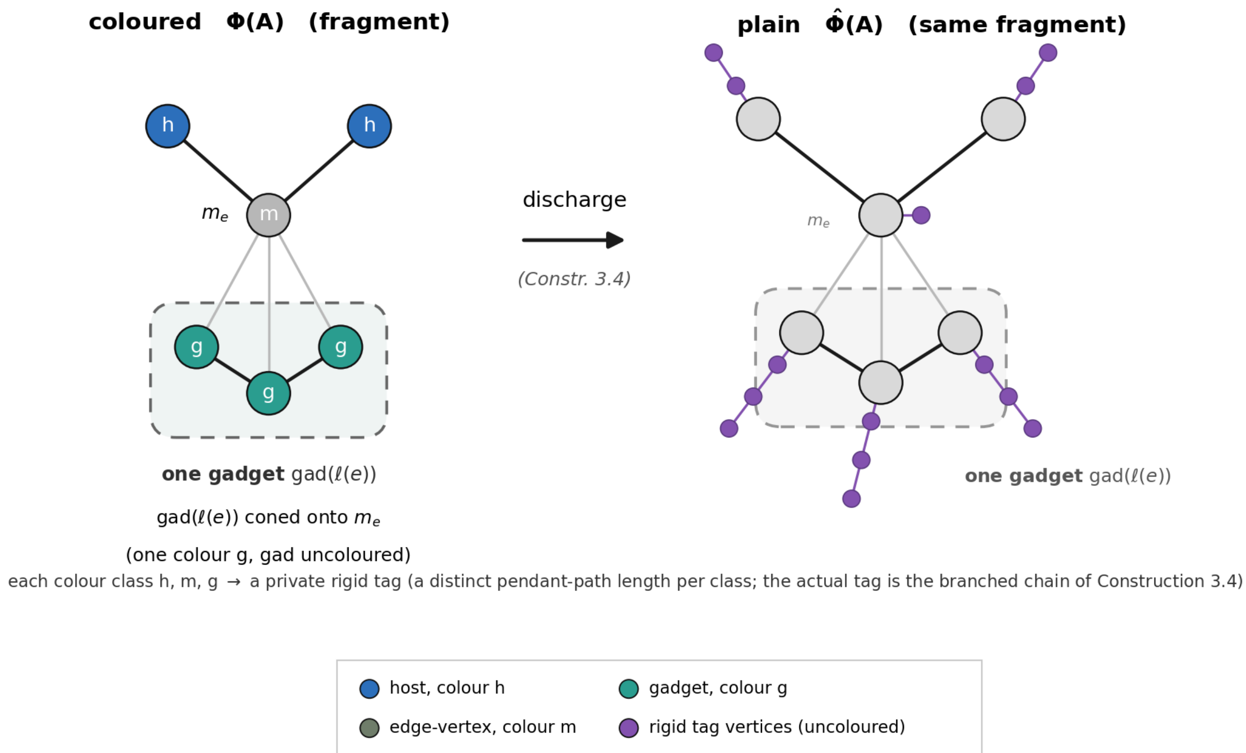

Figure 4. *The colour-discharge of Construction 3.4 on one edge-vertex and its gadget. Each colour class h, m, g is replaced by a private rigid tag (drawn schematically as a distinct pendant path per class; the actual tag is the branched chain of Construction 3.4), so colour-preserving isomorphism of Φ becomes plain isomorphism of $\hat{\Phi}$.*

Theorem 3.5 (BWG-ISO $\equiv$ GI). The map ρ sending (A, B) with $|V_A| = |V_B|$ to ($\hat{\Phi}$(A), $\hat{\Phi}$(B)), and to a fixed non-isomorphic pair when $|V_A| \neq |V_B|$, is a polynomial-time many-one reduction with ρ(A, B) $\in$ GI $\Leftrightarrow$ (A, B) $\in$ BWG-ISO. Together with the trivial reduction GI $\leq^p_m$ BWG-ISO (bare hosts carrying one fixed label on every edge, accepting iff the hosts are isomorphic), BWG-ISO $\equiv^p_m$ GI.

Proof. By Construction 3.4 it suffices to show Φ(A) $\cong_{\text{col}}$ Φ(B) $\Leftrightarrow$ (A, B) $\in$ BWG-ISO. ($\Leftarrow$) Let h witness BWG-ISO. For each e, (ℓ_A(e), ℓ_B(h(e))) $\in$ DSFI gives by Fact 2.3 an isomorphism $g_e : \Gamma_e \to \Gamma_{h(e)}$. Define φ by $\varphi{\restriction}V_A$ = h, $\varphi(m_e) = m_{h(e)}$, and $\varphi{\restriction}$V(Γ_e) = g_e. It is colour-preserving and bijective on each colour block, and preserves all three edge types: incidence u $\sim m_e \mapsto$ h(u) $\sim m_{h(e)}$ since u $\in$ e $\Leftrightarrow$ h(u) $\in$ h(e); cone $m_e \sim$ a $\mapsto m_{h(e)} \sim g_e$(a) since the apex joins all gadget vertices on both sides; gadget a $\sim$ b $\mapsto g_e$(a) $\sim g_e$(b) since g_e is a gadget isomorphism. ($\Rightarrow$) Let $\varphi : \Phi$(A) $\to \Phi$(B) be colour-preserving. Colours force $\varphi(V_A) = V_B$ and $\varphi(\{m_e\}) = \{m_{e'}\}$; set h := $\varphi{\restriction}V_A$. Each m_e has h-neighbourhood exactly e, which in a simple host determines it, so $\varphi(m_e)$ is the m-vertex with h-neighbourhood {h(u), h(v)}; hence {h(u), h(v)} $\in E_B$ and $\varphi(m_e) = m_{h(e)}$. Applying the same to φ^{-1}, h is an incidence-preserving bijection carrying E_A onto E_B, i.e. an isomorphism $H_A \to H_B$. By Lemma 3.3(ii) each $\varphi{\restriction}$V(Γ_e) is a colour-preserving isomorphism $\Gamma_e \to \Gamma_{h(e)}$, whence ($\ell_A$(e), ℓ_B(h(e))) $\in$ DSFI by Fact 2.3, so h witnesses BWG-ISO. For the running time, let s = max$_e$|V(Γ_e)| = poly(max$_e|\ell_A$(e)|); then $|W_A|$ = O(n + ms) and $|F_A|$ = O(ms^2), both polynomial, and the colouring, the three edge types, and the per-colour rigid tags are listed in time linear in these sizes, so ρ is polynomial-time. ■

Remark 3.6 (per-edge reading). For e $\in E_A$, e' $\in E_B$, N_A(e) $\cong N_B$(e') iff there is a host isomorphism h : $H_A \to H_B$ with h(e) = e' and (ℓ_A(f), ℓ_B(h(f))) $\in$ DSFI for every f, by the same argument with $m_e \mapsto m_{e'}$ forced by the individualisation. The matrix [N_A(e) $\cong$ N_B(e')] thus records which label-preserving host isomorphisms route through which edge, while its nonemptiness coincides with the single query Φ(A) $\cong$ Φ(B) of Theorem 3.5.

The construction is specific to the isomorphism variant. It does not transfer to BWG-NI: a polynomial-time GI-reduction of BWG-NI would, by Theorem 7.6(a) below, collapse PH, because BWG-NI is NP-hard whereas GI is low. The non-iso condition cannot be folded into a single GI query in this way, since the displacing witness it requires is not preserved by coning further structure onto the labels.

4. Membership and the hardness profile

Proposition 4.1 (membership). BWG-NI $\in$ NP$^{\text{GI}}$.

Proof. An NP machine with a GI oracle guesses one bijection f, verifies in polynomial time that it is a graph isomorphism of the bare hosts (the host graphs stripped of their labels), then, with gad the label-to-graph map of Fact 2.3, queries the oracle on the pairs gad(ℓ_G(e)), gad(ℓ_H(f(e))) for every edge e of G; by Fact 2.3 a query answers non-isomorphic exactly when (ℓ_G(e), ℓ_H(f(e))) $\notin$ DSFI, which is condition (L^-) of Definition 2.2. It accepts iff all |E| non-adaptive queries answer non-isomorphic. One existential guess followed by GI-oracle queries with a polynomial-time decision is an NP$^{\text{GI}}$ computation. ■

Proposition 4.2 (hardness profile). BWG-NI is GI-hard and GNI-hard, and is not coNP-hard unless PH $= \Sigma_2^{\text{P}}$.

Proof. GI-hardness: label every edge of G with a fixed DNF φ and every edge of H with ψ chosen so (φ, ψ) $\notin$ DSFI; then (L^-) holds for any f, so BWG-NI accepts iff a host isomorphism exists. GNI-hardness: a single-edge instance is (φ, ψ) $\notin$ DSFI, i.e. DSNI, which is GNI-complete by Fact 2.3. Non-coNP-hardness: by Proposition 4.1, BWG-NI $\in$ NP$^{\text{GI}}$, and coNP $\subseteq$ NP$^{\text{GI}}$ already forces PH $= \Sigma_2^{\text{P}}$ (Lemma 6.1); so coNP-hardness of BWG-NI would collapse the hierarchy. The per-edge comparison alone is weaker: DSNI $\equiv$ GNI is coNP-hard iff GI is NP-complete. ■

5. NP-hardness via fixed-point-free automorphisms

Theorem 5.1 (NP-hardness). BWG-NI is NP-hard, via a reduction from the fixed-point-free automorphism problem FPFAUT [Lub81].

Reduction. The reduction maps a graph G to the BWG-NI instance (R(G), R(G)), with R(G) built as follows. We may assume G has no isolated vertices: a single isolated vertex is fixed by every automorphism (map such G to a fixed NO-instance), and if the set I of isolated vertices has |I| $\geq$ 2 then G $\in$ FPFAUT iff G $\setminus$ I $\in$ FPFAUT (extend by any derangement of I), while G = I lies in FPFAUT iff |I| $\geq$ 2; this preprocessing is polynomial. First, subdivide every edge {u, v} of G by a new vertex z_{uv} (the path u–z_{uv}–v), giving S(G); the subdivision lets an automorphism interchange u and v by exchanging the two half-edges {u, z_{uv}} and {z_{uv}, v} rather than fixing the edge (Figure 5). Second, at every original vertex v attach a pinning gadget: a 3-star with centre c and three leaves, the attachment leaf l_1 being also joined to v and the other two, l_2 and l_3, free (Figure 6). Finally, label every edge of R(G) by a distinct DNF, the labels pairwise non-DSFI-isomorphic (Lemma 5.2), and take both hosts equal to this labelled R(G) (Figure 7).

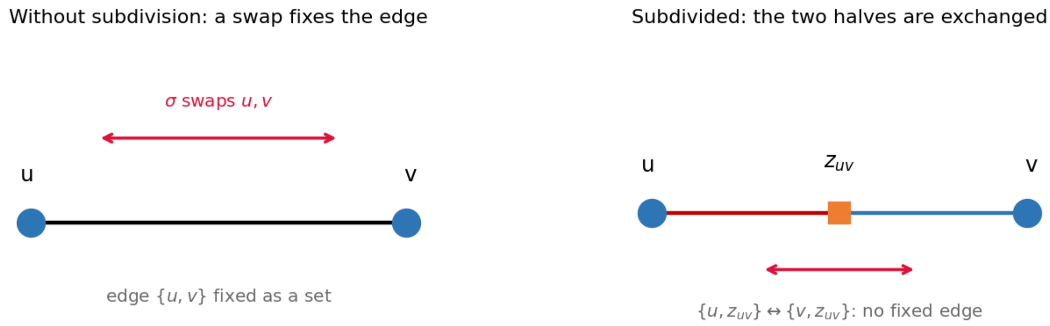

Figure 5. *The subdivision device. Left: an interchange of adjacent endpoints fixes the edge as a set. Right: after inserting z_{uv} the two half-edges are exchanged, so no edge is fixed.*

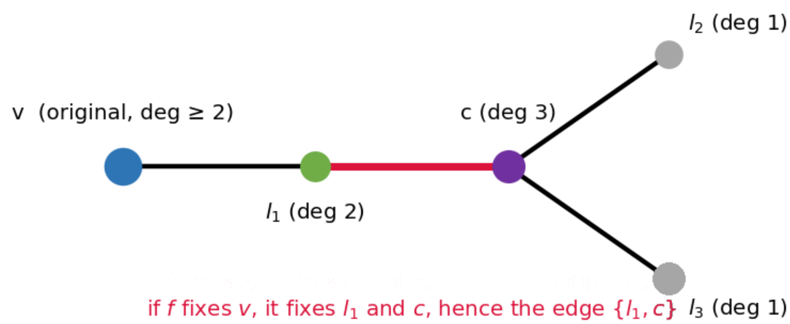

Figure 6. *The pinning gadget at an original vertex v: a 3-star with centre c and leaves l_1, l_2, l_3, attached to v through l_1. The gadget vertices c, l_1, l_2, l_3 have degree multiset (1,1,2,3) in R(G); these degrees together with adjacency to a centre identify every gadget vertex. If an automorphism fixes v it fixes l_1 and c, hence the pinned edge {l_1, c}.*

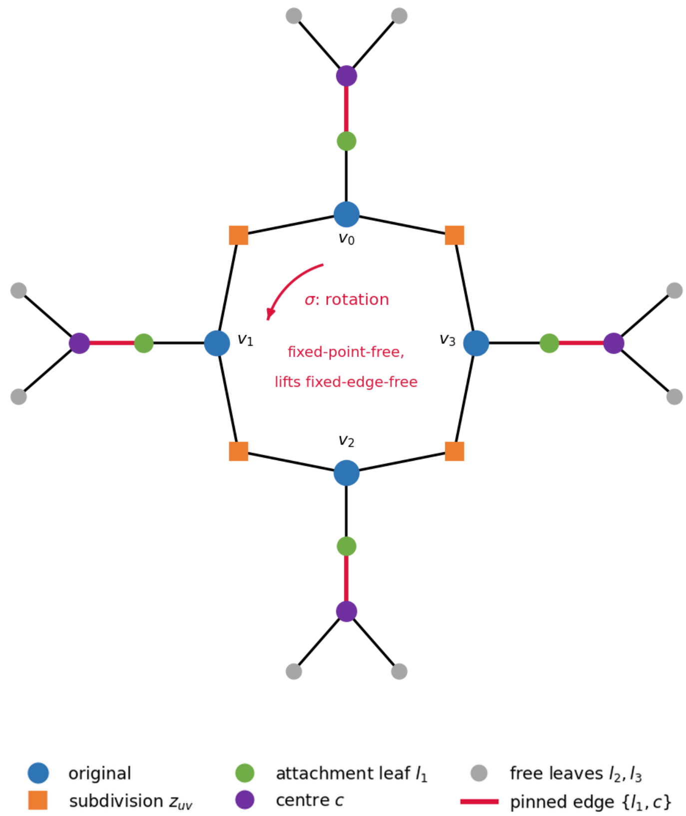

Figure 7. *The construction R(G) for $G = C_4$. The fixed-point-free rotation σ lifts to a fixed-edge-free automorphism of R(G); with all edge labels distinct and pairwise non-DSFI-isomorphic, (L^-) reads off fixed-edge-freeness. Vertices of C_4 are labelled $v_0, \ldots, v_3$ here and $1, \ldots, 4$ in Example 5.5.*

Lemma 5.2 (availability of labels). For every m there is a list $\varphi_1, \ldots, \varphi_m$ of DNF formulas over a common variable set, each of size at most m, pairwise non-DSFI-isomorphic and computable in time polynomial in m.

Proof. Over the variables $z_1, \ldots, z_m$, let φ_i be the DNF consisting of the single monomial $z_1 \wedge \cdots \wedge z_i$. A DSFI-isomorphism renames variables and carries one formula to the other syntactically, hence preserves the multiset of monomial degrees; φ_i has one monomial, of degree i, so $(\varphi_i, \varphi_j) \in$ DSFI forces i = j. The φ_i are pairwise non-DSFI-isomorphic, each of size at most m, and the list is produced in time polynomial in m. ■

Lemma 5.3 (vertex classes and the induced automorphism). Assume G has no isolated vertices and put $n = |V(G)|$. The free leaves (degree 1), the centres (degree 3 with two degree-1 neighbours), the attachment leaves (degree 2 adjacent to a centre), the originals (the non-centres adjacent to an attachment leaf), and the subdivision vertices (the remaining degree-2 vertices) partition V(R(G)), and every automorphism of R(G) preserves these five classes. Each f ∈ Aut(R(G)) restricts on the originals to an automorphism σ of G with $f(z_{uv}) = z_{\sigma(u)\sigma(v)}$, and carries the gadget at v to the gadget at σ(v); conversely every $\sigma \in$ Aut(G) is induced by exactly 2^n automorphisms of R(G), differing by which free leaf maps to which in each of the n gadgets. In particular $|\mathrm{Aut}(R(G))| = 2^n \cdot |\mathrm{Aut}(G)|$.

Class	Degree in R(G)	Identifying test
free leaves l_2, l_3	1	the degree-1 vertices (only neighbour a centre)
centres c	3	exactly two degree-1 neighbours (a degree-3 original has none)
attachment leaves l_1	2	adjacent to a centre (the other neighbour an original)
originals v	$\deg_G(v) + 1 \geq 2$	the non-centres adjacent to an attachment leaf
subdivision vertices z_{uv}	2	the remaining degree-2 vertices (both neighbours originals)

The vertex classification of Lemma 5.3. Degrees are taken in R(G); each test uses the abstract graph alone, and every predicate is isomorphism-invariant.

Proof. In R(G) a free leaf has degree 1 (its only neighbour is a centre); a centre has degree 3 (neighbours the attachment leaf and the two free leaves), two of them of degree 1; an attachment leaf has degree 2 (neighbours v and the centre); an original v has degree $\deg_G$(v) + 1 $\geq$ 2 (its subdivision neighbours together with the attachment leaf); a subdivision vertex z_{uv} has degree 2 (neighbours u and v). The five predicates are therefore mutually exclusive and exhaustive: the degree-1 vertices are exactly the free leaves; among degree-3 vertices a centre is the one with two degree-1 neighbours, whereas an original of G-degree 2 also has degree 3 but all of its neighbours have degree ≥ 2; a degree-2 vertex is an attachment leaf if adjacent to a centre, an original of G-degree 1 if adjacent to an attachment leaf, and a subdivision vertex if its two neighbours are originals; these cases are distinct, since an attachment leaf has one original and one centre as neighbours, a G-degree-1 original has one attachment leaf and one subdivision, and a subdivision has two originals. Across all degrees the originals are exactly the non-centres adjacent to an attachment leaf (an attachment leaf's only neighbours being its original and its centre), and the remaining vertices are the subdivisions. Each predicate is invariant under isomorphism, so every f $\in$ Aut(R(G)) preserves the classes. Hence f permutes the originals; as z_{uv} is the unique subdivision vertex adjacent to both u and v, the restriction σ of f to the originals satisfies f(z_{uv}) $= z_{\sigma(u)\sigma(v)}$ and sends edges of G to edges of G, so $\sigma \in$ Aut(G); and f carries the gadget at v to the gadget at σ(v), taking its centre and attachment leaf to those at σ(v) and its two free leaves onto the two free leaves at σ(v). Conversely any $\sigma \in$ Aut(G) lifts: act by σ on the originals, by $z_{uv} \mapsto z_{\sigma(u)\sigma(v)}$ on the subdivisions, and by the induced map on each gadget, with a free choice in each gadget of the bijection between its two free leaves and those of the image gadget. These n choices are independent and the remainder of f is forced, so exactly 2^n automorphisms induce each σ; summing over σ gives $|$Aut(R(G))$| = 2^n \cdot |$Aut(G)$|$. ■

Lemma 5.4 (fixed edges versus fixed points). For f $\in$ Aut(R(G)) with induced $\sigma \in$ Aut(G), f is fixed-edge-free iff σ is fixed-point-free. Consequently R(G) has a fixed-edge-free automorphism iff G $\in$ FPFAUT.

Proof. If σ(v) = v then f maps the gadget at v to itself and fixes its attachment leaf l_1, the unique gadget vertex adjacent to the fixed v; hence it also fixes the centre c, the unique neighbour of l_1 other than v, so it fixes the edge {l_1, c} and is not fixed-edge-free. Conversely, let σ be fixed-point-free and f any automorphism inducing it; every edge of R(G) is of one of three kinds. A subdivision edge {v, z_{uv}} maps to {σ(v), $z_{\sigma(u)\sigma(v)}$}, which differs from it because σ(v) $\neq$ v and σ(v) is an original while z_{uv} is a subdivision vertex; if σ interchanges the endpoints of {u, v} the two half-edges are merely exchanged. An attachment edge {v, l_1} maps to the attachment edge of the gadget at σ(v) and moves since σ(v) $\neq$ v. An internal gadget edge at v is carried into the gadget at σ(v) $\neq$ v, a different gadget, so it moves, a free-leaf swap notwithstanding. Hence no edge is fixed and f is fixed-edge-free. Ranging over automorphisms and using Lemma 5.3 yields the final equivalence. ■

Proof of Theorem 5.1. The reduction outputs (R(G), R(G)) in polynomial time, its labels by Lemma 5.2. Since the labels are distinct and pairwise non-DSFI-isomorphic, for any host automorphism f and edge e the condition $(\ell(e), \ell(f(e))) \notin$ DSFI holds iff $f(e) \neq e$, so it holds at every edge iff f is fixed-edge-free. By Lemma 5.3 every host automorphism induces $\sigma \in \mathrm{Aut}(G)$, and by Lemma 5.4, R(G) has a fixed-edge-free automorphism iff G $\in$ FPFAUT. Thus (R(G), R(G)) $\in$ BWG-NI iff G $\in$ FPFAUT, and since FPFAUT is NP-complete [Lub81], BWG-NI is NP-hard. The reduction uses general fixed-point-free automorphisms, not only involutions; the subdivision-plus-pinning device is of the kind used by [HSV01] for fixed-edge-free involutory automorphisms. ■

Example 5.5 (the reduction on C_4 and on P$_3$). Take G $= C_4$ with vertices 1, 2, 3, 4 in cyclic order (Figure 8). Then R(C_4) has 24 vertices: the four originals, four subdivision vertices z_{12}, z_{23}, z_{34}, z_{41}, and four pinning gadgets, the i-th with centre c_i, attachment leaf l_1^i, and free leaves l_2^i, l_3^i. The degree multiset locates the classes: free leaves have degree 1, attachment leaves and subdivision vertices degree 2, centres degree 3; and so do the originals, since $\deg_G(v) = 2$ gives each original degree $\deg_G(v) + 1 = 3$. This degree-3 coincidence between originals and centres, the kind of collision that can hide a stray automorphism, is resolved by neighbour degrees exactly as in Lemma 5.3: a centre has two degree-1 neighbours (its free leaves), whereas an original has none, its three neighbours being two subdivision vertices and one attachment leaf, all of degree 2. The rotation $\sigma : i \mapsto i + 1 \pmod 4$ is a fixed-point-free automorphism of C_4. It lifts to f $\in$ Aut(R(C_4)) acting by $i \mapsto i + 1$ on the originals, $z_{i,i+1} \mapsto z_{i+1,i+2}$ on the subdivisions, and carrying the gadget at i to the gadget at i + 1 (any of the 2^4 free-leaf matchings serves). No edge of R(C_4) is fixed: a subdivision edge $\{i, z_{i,i+1}\} \mapsto \{i{+}1, z_{i+1,i+2}\}$, an attachment edge $\{i, l_1^i\} \mapsto \{i{+}1, l_1^{i+1}\}$, and an internal gadget edge at i is carried into the gadget at $i + 1 \neq i$, which is Lemma 5.4 made explicit. Labelling the edges of R(C_4) by the pairwise non-DSFI-isomorphic monomials of Lemma 5.2, every label is displaced, so (L^-) holds at f and (R(C_4), R(C_4)) $\in$ BWG-NI, certifying $C_4 \in$ FPFAUT. Exhaustively, $|\mathrm{Aut}(R(C_4))| = 2^4 \cdot |\mathrm{Aut}(C_4)| = 16 \cdot 8 = 128$, of which $2^4 \cdot$ #FPFAUT(C_4) $= 16 \cdot 5 = 80$ are fixed-edge-free (Lemma 5.3, Remark 5.6). By contrast take G = P$_3$, the path 1–2–3. Its only nontrivial automorphism is the reflection $1 \leftrightarrow 3$, which fixes the centre 2; hence P$_3 \notin$ FPFAUT. In R(P$_3$) every automorphism fixes the original 2, therefore fixes its attachment leaf l_1^2 (the unique gadget vertex adjacent to 2) and then the centre c_2 (the unique neighbour of l_1^2 other than 2), and so fixes the pinned edge $\{l_1^2, c_2\}$: no automorphism is fixed-edge-free, and (R(P$_3$), R(P$_3$)) $\notin$ BWG-NI. The pinning gadget thus turns the fixed vertex 2 into a fixed edge, which is its role in Lemma 5.4.

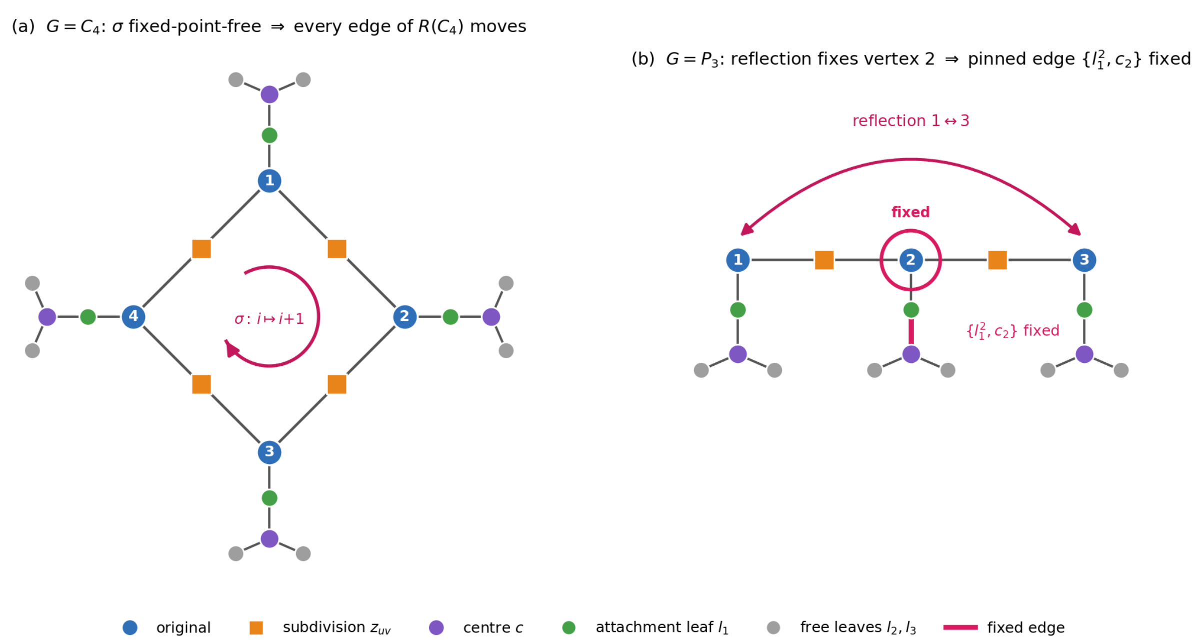

Figure 8. *Example 5.5: the reduction on C_4 (a) and P_3 (b). (a) the rotation $\sigma : i \mapsto i + 1$ is fixed-point-free, and its lift to $R(C_4)$ displaces every edge, so $C_4 \in$ FPFAUT. (b) the reflection of P_3 fixes vertex 2, so every lift fixes the pinned edge $\{l_1^2, c_2\}$ (bold) and $P_3 \notin$ FPFAUT. Vertex classes are coloured as in Figures 5–6; the originals $v_0, \ldots, v_3$ of Figure 7 are the vertices 1, . . . , 4 here.*

Remark 5.6 (counting). By Lemma 5.3 the automorphisms of R(G) inducing a given σ number 2^n, n $= |V(G)|$, and by Lemma 5.4 one of them is fixed-edge-free iff σ is fixed-point-free; hence R(G) has exactly $2^n \cdot$ #FPFAUT(G) fixed-edge-free automorphisms, and these are the accepting witnesses. A reduction is parsimonious if it preserves the number of solutions and weakly parsimonious if the two counts are related by a polynomial-time computable factor; this reduction is weakly parsimonious. An asymmetric replacement gadget would make it exactly parsimonious, but the weak form suffices for Proposition 5.7.

NP-hardness is consistent with membership: NP $\subseteq$ NP$^{\mathrm{GI}}$, so an NP-hard problem may lie in NP$^{\mathrm{GI}}$, unlike a coNP-hard one. The host-automorphism search carries the NP-hardness; the labels only encode the distinctness restriction. The construction uses unbounded-degree hosts; see Section 9.

Proposition 5.7 (counting hardness). #BWG-NI is #P-hard, and #BWG-NI $\in$ #$\cdot$P$^{\mathrm{GI}}$, where #$\cdot$P$^{\mathrm{GI}}$ is the counting operator applied to P$^{\mathrm{GI}}$: the functions $x \mapsto |\{ y : (x, y) \in B \}|$ over polynomially length-bounded witnesses y, for a fixed B $\in$ P$^{\mathrm{GI}}$. Counting reduces to decision for both neighbours of BWG-NI, #GI $\in$ FP$^{\mathrm{GI}}$ [Mat79] and #FI $\in$ FP$^{\mathrm{FI}}$ [AT00, Theorem 5.2], and for the syntactic comparison #DSFI $\in$ FP$^{\mathrm{GI}}$ (DSFI-isomorphisms form a coset of the syntactic automorphism group, so Mathon's pointwise-stabiliser counting [Mat79] applies with a DSFI, hence by Fact 2.3 a GI, oracle [ACL12]; variables are stabilised syntactically by appending each as a singleton term in a distinct multiplicity exceeding the term counts of the input formulas, which every syntactic isomorphism preserves); that #BWG-NI $\in$ FP$^{\mathrm{GI}}$ would instead force PH $= \Sigma_2^{\mathrm{P}}$ makes this a three-way asymmetry, BWG-NI breaking relative to its own natural oracle GI a counting-to-decision reduction that holds for both GI and FI relative to theirs.

Proof. By Remark 5.6, #BWG-NI(R(G), R(G)) $= 2^n \cdot$ #FPFAUT(G); dividing by the computable 2^n gives #FPFAUT $\leq$ #BWG-NI under polynomial-time counting reductions, so #BWG-NI is #P-hard, since #FP-FAUT is #P-complete, by a parsimonious reduction from 3SAT [Lub81, Corollary 2]. The witness predicate uses the GI oracle, so #BWG-NI $\in$ #$\cdot$P$^{\mathrm{GI}}$. If #BWG-NI $\in$ FP$^{\mathrm{GI}}$ then, since #BWG-NI is #P-hard, #P $\subseteq$ FP$^{\mathrm{GI}}$, whence PH $\subseteq$ P$^{\#\mathrm{P}}$ [Tod91] $\subseteq$ P$^{\mathrm{GI}}$ $\subseteq$ Low$_2$ [Sch88], collapsing the hierarchy. ∎

6. Non-completeness and non-equivalence with FI

Lemma 6.1 (Agrawal–Thierauf). If coNP $\subseteq$ NP$^{\mathrm{GI}}$ then PH $= \Sigma_2^{\mathrm{P}}$.

Proof. (i) $\Sigma_2^{\mathrm{P}} \subseteq$ NP$^{\mathrm{GI}}$. Let L $\in \Sigma_2^{\mathrm{P}}$, L $= \{ x : \exists y\ (x, y) \in B \}$ with B $\in$ coNP. By hypothesis B $\in$ NP$^{\mathrm{GI}}$, B $= \{ (x, y) : \exists z\ M^{\mathrm{GI}}(x, y, z)$ accepts $\}$. Then L $= \{ x : \exists (y, z)\ M^{\mathrm{GI}}(x, y, z)$ accepts $\}$, a single existential block over a polynomial-time GI-oracle predicate, hence NP$^{\mathrm{GI}}$. As GI $\in$ NP, NP$^{\mathrm{GI}} \subseteq$ NP$^{\mathrm{NP}} = \Sigma_2^{\mathrm{P}}$, so $\Sigma_2^{\mathrm{P}} =$ NP$^{\mathrm{GI}}$. (ii) $\Sigma_3^{\mathrm{P}} = \mathrm{NP}^{\Sigma_2^{\mathrm{P}}} = \mathrm{NP}^{\mathrm{NP}^{\mathrm{GI}}} = \Sigma_2^{\mathrm{GI}} = \Sigma_2^{\mathrm{P}}$ by GI $\in$ Low$_2$ [Sch88]. Hence PH $= \Sigma_2^{\mathrm{P}}$. ∎

Theorem 6.2 (non-completeness). If BWG-NI is Σ_2^{P}-complete then PH $= \Sigma_2^{\mathrm{P}}$.

Proof. If BWG-NI is Σ_2^{P}-complete then $\Sigma_2^{\mathrm{P}} \leq_m^p$ BWG-NI $\in$ NP$^{\mathrm{GI}}$; as NP$^{\mathrm{GI}}$ is $\leq_m^p$-closed (compose the reduction with the oracle machine), $\Sigma_2^{\mathrm{P}} \subseteq$ NP$^{\mathrm{GI}}$, so coNP $\subseteq$ NP$^{\mathrm{GI}}$ and PH $= \Sigma_2^{\mathrm{P}}$ by Lemma 6.1. ∎

Corollary 6.3. No coNP-hard problem lies in NP$^{\mathrm{GI}}$ unless PH $= \Sigma_2^{\mathrm{P}}$. BWG-NI is non-complete as a non-coNP-hard inhabitant of NP$^{\mathrm{GI}}$.

Proof. A coNP-hard L $\in$ NP$^{\mathrm{GI}}$ would place coNP $\subseteq$ NP$^{\mathrm{GI}}$ by the $\leq_m^p$-closure of NP$^{\mathrm{GI}}$, whence PH $= \Sigma_2^{\mathrm{P}}$ by Lemma 6.1. ■

Theorem 6.4 (non-equivalence). If BWG-NI $\equiv_m^p$ FI then PH $= \Sigma_2^{\mathrm{P}}$. Already the single direction FI $\leq_m^p$ BWG-NI implies coNP $\subseteq$ NP$^{\mathrm{GI}}$, hence PH $= \Sigma_2^{\mathrm{P}}$.

Proof. FI is coNP-hard [AT00]. FI $\leq_m^p$ BWG-NI would make BWG-NI coNP-hard; with BWG-NI $\in$ NP$^{\mathrm{GI}}$ and NP$^{\mathrm{GI}}$ $\leq_m^p$-closed, coNP $\subseteq$ NP$^{\mathrm{GI}}$, so PH $= \Sigma_2^{\mathrm{P}}$ by Lemma 6.1. ■

The mechanism is whether the per-edge comparison is an isomorphism test or its negation. The factorisation FI $= \exists\lambda\cdot$EQ has a coNP-complete inner test, making FI coNP-hard; BWG-NI never certifies a full coNP fact, its per-edge comparison being the negation of a GI-confined test (DSNI $\equiv$ GNI), so the per-edge comparison alone is only GNI-hard. Section 7 makes this quantitative.

Remark 6.5 (unconditional versus randomized NP-hardness). Whether FI is NP-hard is a challenging open question [AT00]; the hardest NP problem known to reduce to it is GI [AT00], so the established deterministic NP-hardness floor of FI reaches only the GI degree. BWG-NI is NP-hard unconditionally (Theorem 5.1), so its floor is all of NP. FI is NP-hard under randomized reductions only by composition, from SAT $\leq_r$ USAT (unique satisfiability) [VV86], USAT $\leq_m^p$ FC [AT00], and FC $\equiv_m^p$ FI [BRS98], FC being formula congruence (a variable permutation composed with negations of variables); [AT00] state the randomized hardness for USAT, not as a theorem about FI. The remaining points of difference are coNP-hardness (FI yes, BWG-NI no) and NP$^{\mathrm{GI}}$ membership (BWG-NI yes, FI not unless the hierarchy collapses).

7. Degree, lowness, and the relativized separation

This section locates the source of the complexity: the single feature separating the NP-hard general problem from its tractable rigid restriction (hosts with no nontrivial automorphism) is the quantifier over host isomorphisms, the labels contributing only a polynomial-time distinctness condition. On rigid hosts the problem is captured by the conjunctive closure of GNI (Theorem 7.2) and is low for Σ_2^{P} (Theorem 7.4); Proposition 7.5 records that this quantifier alone accounts for each separation between the rigid and general cases. Throughout, BWG-NI$_{\mathrm{rig}}$ denotes the restriction of BWG-NI to instances in which both host graphs are rigid.

Definition 7.1 (conjunctive closure of GNI). CGNI$^{\wedge}$ is the class of L for which a polynomial-time map sends each x to graph pairs $(A_1, B_1), \ldots, (A_m, B_m)$ with x $\in$ L $\Leftrightarrow \bigwedge_i$ GNI(A_i, B_i). A problem is OR-closed (AND-closed) if the disjunction (conjunction) of any polynomially many of its instances reduces to a single instance. GNI is OR-closed (a disjunction of non-isomorphisms is one GNI instance, via disjoint unions whose components are individualised so that any isomorphism matches the i-th components), and it is also AND-closed, since an AND of non-isomorphisms is the complement of an OR of isomorphisms: GI has a polynomial-time computable or-function [KST93, Theorem 1.42] (originally [Cha89, LT92]), built from the ordered-pair encoding in the proof of [KST93, Theorem 1.40] together with the unordered-pair behaviour of disjoint unions of connected graphs [KST93, Fact 1.41], so by complementation GNI has a polynomial-time computable and-function. Hence CGNI$^{\wedge}$ coincides with the $\leq_m^p$-degree of GNI, and the CGNI$^{\wedge}$-completeness below is GNI-completeness.

Theorem 7.2 (degree). CGNI$^{\wedge}$ $\leq_m^p$ BWG-NI. On arbitrary rigid hosts BWG-NI$_{\mathrm{rig}}$ $\in$ P$^{\mathrm{GI}}$. On rigid hosts drawn from a polynomial-time-recognizable class on which an isomorphism is computable, and triviality of the automorphism group decidable, in polynomial time (for instance rigid trees), BWG-NI$_{\mathrm{rig}}$ is moreover $\leq_m^p$-complete for CGNI$^{\wedge}$, equivalently for GNI.

Proof. Hardness. Given $\bigwedge_i \mathrm{GNI}(A_i, B_i)$, i = 1, …, m, first pad with fixed true conjuncts (each a fixed non-isomorphic pair) so that m $\geq$ 3. Let R be the tree obtained from the path u_0–u_1–$\cdots$–u_m by attaching at u_m two pendant paths (the markers), of lengths one and two. Then u_m is the unique vertex of degree three and its three branches have distinct lengths m, 1, 2, so R is rigid (a path graph, by contrast, retains the reversal automorphism); the unique isomorphism between two copies of R is the identity. Take both hosts equal to R; label the i-th path edge $\{u_{i-1}, u_i\}$ with dnf(A_i) in G and dnf(B_i) in H, where dnf is the GI $\leq_m^p$ DSFI reduction provided by Fact 2.3 (so $A_i \cong B_i$ iff (dnf(A_i), dnf(B_i)) $\in$ DSFI), and label every marker edge of G with one fixed DNF and every marker edge of H with a fixed non-DSFI-isomorphic partner. The unique host isomorphism is the identity, pairing edge i with edge i and marker edges with marker edges; the marker labels always disagree, so (L^-) holds iff every path edge disagrees, i.e. iff $\bigwedge_i \mathrm{GNI}(A_i, B_i)$. The image lies in BWG-NI$_{\mathrm{rig}}$ and is a rigid tree. Membership. The machine first rejects unless both hosts are rigid; rigidity is decidable in $\mathrm{P^{GI}}$ since GA, the problem of deciding whether a graph has a nontrivial automorphism, is Turing-reducible to GI [KST93, Example 1.10]. On arbitrary rigid hosts the unique isomorphism f, or the fact that none exists, is computable in $\mathrm{FP^{GI}}$: search reduces to decision for GI by individualisation [KST93], fixing f vertex by vertex in O(n^2) oracle calls. A $\mathrm{P^{GI}}$ machine computes f and then decides the per-edge disequations (ℓ(e), ℓ(f(e))) $\notin$ DSFI with $|E|$ further GI queries (each DSFI test is a GI query by Fact 2.3), so BWG-NI$_{\mathrm{rig}}$ $\in \mathrm{P^{GI}}$ with no tractability assumption. When the rigid hosts are moreover drawn from such a class, the map first checks in polynomial time that both hosts lie in the class and are rigid, outputting a fixed isomorphic pair (a false conjunct) otherwise; f is then computable without an oracle in polynomial time, and if none exists the map outputs the same false conjunct; otherwise output the gadget pairs (ℓ(e), ℓ(f(e))) over all edges e, so acceptance is $\bigwedge_e \mathrm{GNI}(\ell(e), \ell(f(e)))$ and BWG-NI$_{\mathrm{rig}}$ $\in$ CGNI$^{\wedge}$. ■

Canonicity reaches the same degree by a different route: the complement of the canonical restriction of the semantic comparison is GNI-complete, CFNI = co-CFI $\equiv_m^p$ GNI [Cri26, Theorem 3]; rigidity removes the host quantifier, canonicity the representation ambiguity, and both land on the GNI degree.

Proposition 7.3 (the degree lies in the low hierarchy). GNI $\subseteq$ CGNI$^{\wedge}$ $\subseteq \mathrm{P^{GI}} \subseteq \mathrm{Low}_2 \subseteq \Sigma_2^{\mathrm{P}}$.

Proof. GNI is decided by one GI query; CGNI$^{\wedge}$ by m non-adaptive queries, so CGNI$^{\wedge}$ $\subseteq \mathrm{P^{GI}}$. Any A $\in \mathrm{P^{GI}}$ satisfies $\Sigma_2^A \subseteq \Sigma_2^{\mathrm{GI}} = \Sigma_2^{\mathrm{P}}$ [Sch88], i.e. A $\in \mathrm{Low}_2$. ■

The third inclusion is the operative one. CGNI$^{\wedge}$ $\subseteq \mathrm{Low}_2$, whereas coNP $\subseteq \mathrm{Low}_2$ would force $\Sigma_3^{\mathrm{P}} = \Sigma_2^{\mathrm{P}}$ (since $\Sigma_2^{\mathrm{coNP}} = \Sigma_3^{\mathrm{P}}$). So unless the hierarchy collapses, the hardness floor of BWG-NI sits inside the low part of Σ_2^{P}, not in the coNP-complete part where the floor of FI sits.

Theorem 7.4 (rigid lowness). BWG-NI$_{\mathrm{rig}}$ is low for Σ_2^{P}; in particular it is not Σ_2^{P}-complete unless PH = Σ_2^{P}.

Proof. BWG-NI$_{\mathrm{rig}}$ is a decision problem (a language, not a promise problem: rigidity of the hosts is decidable in $\mathrm{P^{GI}}$ since GA is Turing-reducible to GI [KST93, Example 1.10], so ill-formed instances are rejected rather than left unpromised), and BWG-NI$_{\mathrm{rig}}$ $\in \mathrm{P^{GI}} \subseteq \mathrm{Low}_2$ for arbitrary rigid hosts (Theorem 7.2, Proposition 7.3). The Sch88 lowness $\Sigma_2^L = \Sigma_2^{\mathrm{P}}$ therefore applies to it as a set; were it Σ_2^{P}-complete, $\Sigma_3^{\mathrm{P}} = \Sigma_2^{\mathrm{P}}$. ■

Proposition 7.5 (the host-isomorphism quantifier accounts for the rigid/general distinction). BWG-NI and BWG-NI$_{\mathrm{rig}}$ differ only in the quantifier over host isomorphisms. In BWG-NI the witness f ranges over all isomorphisms G $\to$ H, a coset of Aut(G), under the per-edge disequation predicate, which is not Aut(G)-invariant; in BWG-NI$_{\mathrm{rig}}$ the host is rigid, so this coset is a single map and the quantifier is trivial. This quantifier is the sole source of each complexity difference established above: (i) on rigid hosts drawn from a class as in Theorem 7.2, BWG-NI$_{\mathrm{rig}}$ is $\leq_m^p$-complete for CGNI$^{\wedge}$ (Theorem 7.2), whereas BWG-NI is CGNI$^{\wedge}$-hard but does not lie in CGNI$^{\wedge}$ unless PH = Σ_2^{P}; (ii) BWG-NI$_{\mathrm{rig}}$ $\in \mathrm{P^{GI}}$ (Theorem 7.2), whereas BWG-NI $\in \mathrm{NP^{GI}}$ and BWG-NI $\notin \mathrm{P^{GI}}$ unless PH = Σ_2^{P} (Theorem 7.6(a), Corollary 7.7); (iii) BWG-NI$_{\mathrm{rig}}$ $\in$

Low_2 (Theorem 7.4), whereas BWG-NI $\notin \mathrm{Low}_2$ unless PH $= \Sigma_2^{\mathrm{P}}$ (Theorem 7.6(c)); (iv) co-BWG-NI $\in \Pi_2^{\mathrm{P}}$ natively, and BWG-NI $\in \mathrm{coNP}^{\mathrm{GI}}$ (equivalently co-BWG-NI $\in \mathrm{NP}^{\mathrm{GI}}$) implies PH $= \Sigma_2^{\mathrm{P}}$ (Theorem 7.6(b)). When $\mathrm{Aut(G)} = 1$ each statement reduces to its BWG-$\mathrm{NI}_{\mathrm{rig}}$ form.

Proof. Immediate: the two problems have identical definitions except that the host bijection is quantified over all isomorphisms $\mathrm{G} \to \mathrm{H}$ in BWG-NI and is fixed to the unique such isomorphism in BWG-$\mathrm{NI}_{\mathrm{rig}}$; (i)–(iv) are the cited results. ■

Theorem 7.6 (NP-hardness excludes P^{GI}, $\mathrm{coNP}^{\mathrm{GI}}$, and Low_2). Each of the following implies PH $= \Sigma_2^{\mathrm{P}}$: (a) BWG-NI $\in \mathrm{P}^{\mathrm{GI}}$; (b) BWG-NI $\in \mathrm{coNP}^{\mathrm{GI}}$; (c) BWG-NI $\in \mathrm{Low}_2$. Moreover (a) also yields NP $\subseteq$ SPP. Hence, unless PH $= \Sigma_2^{\mathrm{P}}$, general BWG-NI lies in none of P^{GI}, $\mathrm{coNP}^{\mathrm{GI}}$, Low_2.

Proof. BWG-NI is NP-hard (Theorem 5.1). (a) BWG-NI $\in \mathrm{P}^{\mathrm{GI}}$ gives NP $\subseteq \mathrm{P}^{\mathrm{GI}} \subseteq \mathrm{Low}_2$, so $\Sigma_2^{\mathrm{SAT}} = \Sigma_2^{\mathrm{P}}$; but $\Sigma_2^{\mathrm{SAT}} = \Sigma_3^{\mathrm{P}}$, so $\Sigma_3^{\mathrm{P}} = \Sigma_2^{\mathrm{P}}$. For the SPP consequence, GI $\in$ SPP [AK02] and SPP is self-low, $\mathrm{P}^{\mathrm{SPP}} = \mathrm{SPP}$ [FFK94], so NP $\subseteq \mathrm{P}^{\mathrm{GI}} \subseteq \mathrm{P}^{\mathrm{SPP}} = \mathrm{SPP}$. (b) BWG-NI $\in \mathrm{coNP}^{\mathrm{GI}}$ gives coNP $\subseteq \mathrm{NP}^{\mathrm{GI}}$, so PH $= \Sigma_2^{\mathrm{P}}$ by Lemma 6.1. (c) NP-hardness gives $\Sigma_2^{\text{BWG-NI}} \supseteq \Sigma_2^{\mathrm{SAT}} = \Sigma_3^{\mathrm{P}}$; if BWG-NI $\in \mathrm{Low}_2$ then $\Sigma_3^{\mathrm{P}} = \Sigma_2^{\mathrm{P}}$. ■

Corollary 7.7 (a witness to a relativized separation). If PH is infinite then $\mathrm{P}^{\mathrm{GI}} \neq \mathrm{NP}^{\mathrm{GI}}$, and BWG-NI witnesses the separation: $\mathrm{P}^{\mathrm{GI}} = \mathrm{NP}^{\mathrm{GI}} \Rightarrow \mathrm{NP} \subseteq \mathrm{P}^{\mathrm{GI}} \subseteq \mathrm{Low}_2 \Rightarrow \mathrm{PH} = \Sigma_2^{\mathrm{P}}$, so under infinite PH, BWG-NI $\in \mathrm{NP}^{\mathrm{GI}} \setminus \mathrm{P}^{\mathrm{GI}}$.

The collapse needs no witness problem: NP $\subseteq \mathrm{NP}^{\mathrm{GI}}$ always, so $\mathrm{NP}^{\mathrm{GI}} = \mathrm{P}^{\mathrm{GI}}$ gives NP $\subseteq \mathrm{P}^{\mathrm{GI}}$, and $\mathrm{P}^{\mathrm{GI}} \subseteq \mathrm{Low}_2$ [Sch88] yields PH $= \Sigma_2^{\mathrm{P}}$. The role of BWG-NI is to provide a natural problem in $\mathrm{NP}^{\mathrm{GI}} \setminus \mathrm{P}^{\mathrm{GI}}$: under PH $\neq \Sigma_2^{\mathrm{P}}$ it lies there, with Theorem 7.6(a) adding NP $\subseteq$ SPP to the cost of the contrary. By the relativizing nature of the arguments this is conditional and does not yield an unconditional separation. A complementary barrier holds at the witness level: BWG-NI $\notin \mathrm{P}^{\mathrm{GI}}$ already implies P $\neq$ NP, since P $=$ NP gives PH $=$ P and then BWG-NI $\in \mathrm{NP}^{\mathrm{GI}} \subseteq \Sigma_2^{\mathrm{P}} = \mathrm{P} \subseteq \mathrm{P}^{\mathrm{GI}}$; an unconditional proof of the exclusion is therefore at least as hard as separating P from NP.

Remark 7.8 (hardness floor). The hardness floor of BWG-NI is the join of two independent sources: a label part (GNI and $\mathrm{CGNI}^{\wedge}$, inside $\mathrm{P}^{\mathrm{GI}} \subseteq \mathrm{Low}_2$) and a search part (the NP-hardness, outside Low_2 unless collapse). Rigidity removes the search floor and leaves only the label floor. More generally, a host restriction lowers BWG-NI into P^{GI} when it removes the search part, that is, when the per-edge disequations no longer couple across the host-automorphism action; triviality of the host automorphism group is one sufficient condition for this decoupling but not evidently the only one, since the search part requires only that the disequation constraints separate, not that the automorphism coset reduce to a single point.

8. Label alphabet and the complexity landscape

The per-edge comparison of BWG-NI is matrix isomorphism over a bounded integer alphabet, which is GI-complete by Fact 2.3; the monotone fragment is already complete and binary matrices suffice [ACL12]. One refinement follows.

Corollary 8.1 (monotone, two-symbol labels suffice). BWG-NI restricted to monotone-DNF edge labels, equivalently {0,1}-matrix labels, remains NP-hard, GNI-hard, in $\mathrm{NP}^{\mathrm{GI}}$, and not Σ_2^{P}-complete unless PH $= \Sigma_2^{\mathrm{P}}$.

Proof. The single-edge case is the negation of monotone DSFI and is GNI-complete since the monotone fragment is GI-complete [ACL12]. Membership is unchanged. The reduction of Theorem 5.1 already uses monotone labels: the single-monomial DNFs of Lemma 5.2 are pairwise non-DSFI-isomorphic, monotone, and polynomial-time constructible, so its NP-hardness holds verbatim for monotone-DNF labels, equivalently {0,1}-matrices. Non-Σ_2^P-completeness follows from NP^{GI} membership as in Theorem 6.2. ■

Collapse level. If BWG-NI is Σ_2^P-complete, Lemma 6.1(i) gives $\Sigma_2^P = NP^{GI}$ and (ii) gives $PH = \Sigma_2^P$; no deeper collapse is evident: $\Sigma_2^P = NP^{GI}$ is not known to imply NP = coNP. Thus the collapse level obtained is Σ_2^P, one level below the level forced by Σ_2^P-completeness of FI. Inside the formula-isomorphism family the same sharpening from the third level to the second is achieved by the canonical restriction [Cri26, Theorem 7]. Agrawal and Thierauf [AT00] show FI is not Σ_2^P-complete unless PH collapses, via a one-round interactive proof for the complementary problem FNI with an NP oracle, i.e. FNI $\in AM^{NP} = BP\cdot\Sigma_2^P$; this places the FI collapse at the third level, by the relativised Boppana–Håstad–Zachos argument [BHZ87]. The second-level collapse available here is a consequence of BWG-NI $\in NP^{GI}$: FI is coNP-hard (Fact 2.4) and so lies outside NP^{GI} unless $PH = \Sigma_2^P$ (Lemma 6.1), leaving the GI-oracle route unavailable to it.

Complement asymmetry. In the complement every bijection must either fail to be a host isomorphism or match some edge pair of syntactically isomorphic labels; formally co-BWG-NI = ∀f [f a graph isomorphism ⇒ ∃e (ℓ_G(e), ℓ_H(f(e))) ∈ DSFI]. The inner comparison is positive and guessable (DSFI ∈ NP), so co-BWG-NI $\in \Pi_2^P$ natively. The GI oracle is used only on the BWG-NI side.

NP-completeness under GI ∈ coNP. BWG-NI is in NP iff GNI ∈ NP, i.e. GI ∈ coNP: the forward direction is the GNI-hardness of Proposition 4.2 (GNI $\leq_m^p$ BWG-NI, and NP is closed under $\leq_m^p$), and for the converse, if GNI ∈ NP an NP machine guesses the host bijection f and, for each edge, a witness that ℓ(e) and ℓ(f(e)) are non-DSFI-isomorphic, each such test being DSNI and hence GNI-complete. Under that hypothesis BWG-NI is NP-complete. The canonical restriction of the semantic comparison meets the same equivalence: CFI ∈ NP ∩ coNP iff GI ∈ coNP, equivalently $NP^{GI} = NP$ [Cri26, Theorem 8]. GI ∈ coNP follows from AM = NP (via GNI ∈ AM) and a fortiori from GI ∈ P. Each such hypothesis places BWG-NI ∈ NP, hence it is NP-complete; the label-disequality variant is NP-complete with no hypothesis (Remark 8.2). The unconditional statement is: NP-hard, in NP^{GI}, not coNP-hard or Σ_2^P-complete unless $PH = \Sigma_2^P$. Under GI ∈ P, $NP^{GI} = NP$ and Theorem 7.6 reads “BWG-NI ∈ P ⇒ P = NP”.

Figure 9 records the landscape under the hypothesis $PH \neq \Sigma_2^P$. BWG-NI lies in NP^{GI} but outside P^{GI}, Low_2, and coNP (Theorems 5.1, 6.2, 7.6), and outside NP when in addition GI ∉ coNP (BWG-NI ∈ NP ⇔ GI ∈ coNP); FI lies in Σ_2^P but outside NP^{GI}, its coNP-hardness being what excludes it (Theorem 6.4, Lemma 6.1); BWG-ISO sits with GI and GNI inside $P^{GI} \subseteq Low_2$ (Theorem 3.5). FACT, the decision form of integer factorisation (given n and k in binary, whether n has a nontrivial factor at most k), lies in NP ∩ coNP and appears for orientation: FACT $\leq_m^p$ BWG-NI by NP-hardness, while the converse reduction would give NP = coNP. Under GI ∈ coNP, BWG-NI drops into NP and is NP-complete.

Remark 8.2 (string comparator; unconditional NP-completeness). With labels compared by plain string disequality, a per-edge test in P, BWG-NI is NP-complete with no hypothesis: membership needs no oracle, since a machine guesses the host bijection and checks the per-edge string comparisons in polynomial time, and the reduction of Theorem 5.1 applies verbatim, its single-monomial labels being pairwise distinct as strings, so identical labels compare equal and distinct labels compare unequal under either comparator.

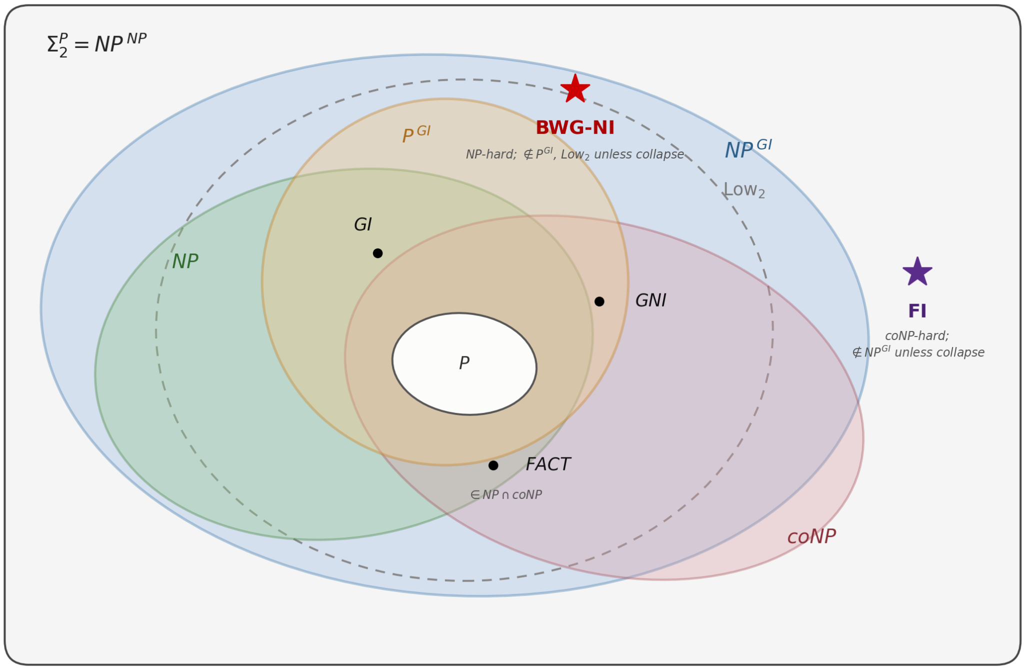

Figure 9. *The complexity landscape under $PH \neq \Sigma_2^{\mathrm{P}}$; the placement of BWG-NI (star) outside NP additionally assumes GI $\notin$ coNP, since BWG-NI $\in$ NP $\Leftrightarrow$ GI $\in$ coNP (Section 8). BWG-NI lies in* $\mathrm{NP}^{\mathrm{GI}} \setminus (\mathrm{P}^{\mathrm{GI}} \cup$ *NP* $\cup$ *coNP); FI lies in* $\Sigma_2^{\mathrm{P}} \setminus \mathrm{NP}^{\mathrm{GI}}$*; GI, GNI, and BWG-ISO lie in* $\mathrm{P}^{\mathrm{GI}} \subseteq \mathrm{Low}_2$*. FACT $\in$ NP $\cap$ coNP is shown for orientation. Positions not implied by the stated results are conjectural.*

9. Caveats and open problems

Provenance and verification. Among the results above, the one resting on a construction rather than a composition with classical facts is the NP-hardness (Theorem 5.1); its consequences (the exclusions from P^{GI}, $\mathrm{coNP}^{\mathrm{GI}}$, and Low_2 by Theorem 7.6, the rigid/general distinction, the witness to $\mathrm{P}^{\mathrm{GI}} \neq \mathrm{NP}^{\mathrm{GI}}$ by Corollary 7.7, and the counting hardness by Proposition 5.7) follow from it together with [Lub81, Sch88, AK02, FFK94, Tod91]. The isomorphism-variant reduction (Theorem 3.5) is a second construction. The membership (Proposition 4.1), Lemma 6.1, and Proposition 7.3 are compositions of standard arguments. Three caveats. (i) The NP-hardness rests on the reduction of Theorem 5.1, which is new relative to the literature and should be independently verified; it uses unbounded-degree hosts deliberately, since a bounded-degree reduction risks vacuity, FPFAUT being possibly tractable on bounded-degree graphs, consistent with [Lub81]'s hard instances being unbounded-degree. (ii) The conjunctive closure $\mathrm{CGNI}^{\wedge}$ coincides with the $\leq_m^p$-degree of GNI by [KST93, Theorem 1.42] (Definition 7.1), so the rigid-host completeness and lowness are GNI-level results and the conjunctive framing is retained only as notation; its relation to the bounded-query-over-GI literature [Wag90, AK00, LT92] is expository. (iii) Fact 2.3 is used under a fixed reading of the attachment clause of [ACL12], which says only that each entry-vertex is linked to the clique codifying its value: here each entry-vertex joins a designated vertex of that clique by a single edge. Under this reading, graph isomorphism of the resulting gadgets and syntactic DNF isomorphism induce the same partition on all 45,759 DNFs with at most three variables and at most three term-occurrences, repeated terms included (checked exhaustively), so the concrete description in Fact 2.3 rests on [ACL12, Theorem 2] together with that verification. The attribution of the subdivision-plus-pinning device to [HSV01] is given as a comparison: that paper states its constructions for involutory fixed-edge-free automorphisms, whereas the present reduction uses general fixed-point-free automorphisms.

Tractable host restrictions. Rigidity places BWG-NI in P^{GI} by trivialising the host-automorphism search (Theorem 7.2). Any host restriction with the same effect is necessarily non-generic: if restricting the host to some class keeps BWG-NI in P^{GI}, then a many-one reduction of general BWG-NI to that restricted problem would place an NP-hard problem in $P^{GI} \subseteq \mathrm{Low}_2$ (Theorem 5.1, [Sch88]) and collapse PH, so no tractable host restriction simulates the general problem under $\leq_m^p$ unless the hierarchy collapses. It is open which structural parameter of the host governs membership in P^{GI} (equivalently, which parameter controls whether the per-edge disequations decouple across the host-automorphism action) and how far beyond rigidity tractability extends.

Bounded-degree host-automorphism search. The NP-hardness reduction uses unbounded-degree hosts. Whether BWG-NI restricted to bounded-degree hosts is NP-hard depends on whether FPFAUT is NP-hard on bounded-degree graphs, a question about the host-automorphism search to which the group-theoretic machinery for bounded-valence isomorphism [Luk82] applies but which is not settled here.

Precomputable versus full oracle access. Whether $NP^{GI}_{pre} = NP^{GI}$ (whether host-automorphism structure can combine precomputed GI facts into genuinely new queries) is open; cardinality arguments and Weisfeiler–Leman/CFI constructions each fail to separate the two (not shown here). BWG-NI lies in NP^{GI}_{pre}.

10. References

[ACL12] G. Ausiello, F. Cristiano, and L. Laura. Syntactic Isomorphism of CNF Boolean Formulas is Graph Isomorphism Complete. ECCC, Report No. TR12-122, 2012.

[AK00] V. Arvind and J. Köbler. Graph isomorphism is low for ZPP(NP) and other lowness results. Proc. STACS 2000, LNCS 1770, Springer, 2000.

[AK02] V. Arvind and P. P. Kurur. Graph Isomorphism is in SPP. Proc. FOCS 2002; Information and Computation 204(5):835–852, 2006.

[AT00] M. Agrawal and T. Thierauf. The formula isomorphism problem. SIAM J. Comput. 30(3):990–1009, 2000.

[BHZ87] R. Boppana, J. Håstad, and S. Zachos. Does co-NP have short interactive proofs? Inform. Process. Lett. 25:127–132, 1987.

[BR93] B. Borchert and D. Ranjan. The Subfunction Relations are Σ_2^P-Complete. Technical Report MPI-I-93-121, Max-Planck-Institut für Informatik, Saarbrücken, 1993.

[BRS98] B. Borchert, D. Ranjan, and F. Stephan. On the computational complexity of some classical equivalence relations on Boolean functions. Theory Comput. Systems 31(6):679–693, 1998.

[Cha89] R. Chang. On the structure of bounded queries to arbitrary NP sets. Proc. 4th Structure in Complexity Theory Conf., IEEE, 1989, 250–258.

[Cri26] F. Cristiano. The complexity landscape of Boolean formula isomorphism: from graph isomorphism to the second level. Electronic Colloquium on Computational Complexity (ECCC), Report TR26-114, 2026.

[FFK94] S. Fenner, L. Fortnow, and S. Kurtz. Gap-definable counting classes. J. Comput. System Sci. 48(1):116–148, 1994.

[GMW91] O. Goldreich, S. Micali, and A. Wigderson. Proofs that yield nothing but their validity or all languages in NP have zero-knowledge proof systems. J. ACM 38(3):690–728, 1991.

[HSV01] F. Harary, W. Slany, and O. Verbitsky. A Symmetric Strategy in Graph Avoidance Games. arXiv:cs/0110049; DBAI-TR-2001-42, TU Wien, 2001. Also in: More Games of No Chance, MSRI Publications 42, Cambridge University Press, 2002.

[KST93] J. Köbler, U. Schöning, and J. Torán. The Graph Isomorphism Problem: Its Structural Complexity. Birkhäuser, Boston, 1993.

[LT92] A. Lozano and J. Torán. On the nonuniform complexity of the graph isomorphism problem. Proc. Structure in Complexity Theory, 1992.

[Lub81] A. Lubiw. Some NP-complete problems similar to graph isomorphism. SIAM J. Comput. 10(1):11–21, 1981.

[Luk82] E. M. Luks. Isomorphism of graphs of bounded valence can be tested in polynomial time. J. Comput. System Sci. 25(1):42–65, 1982.

[Mat79] R. Mathon. A note on the graph isomorphism counting problem. Inform. Process. Lett. 8(3):131–132, 1979.

[Sch88] U. Schöning. Graph isomorphism is in the low hierarchy. J. Comput. System Sci. 37:312–323, 1988.

[Tod91] S. Toda. PP is as hard as the polynomial-time hierarchy. SIAM J. Comput. 20(5):865–877, 1991.

[VV86] L. Valiant and V. Vazirani. NP is as easy as detecting unique solutions. Theoret. Comput. Sci. 47:85–93, 1986.

[Wag90] K. W. Wagner. Bounded query classes. SIAM J. Comput. 19(5):833–846, 1990.

ECCC ISSN 1433-8092

https://eccc.weizmann.ac.il