A Dichotomy for Constrained Eulerian Orientations

Jincheng Guan guanjincheng@mail.ustc.edu.cn
Shuai Shao shao10@ustc.edu.cn
Ke Shi self.ke.shi@gmail.com

School of Computer Science and Technology & Hefei National Laboratory
University of Science and Technology of China

Abstract

We establish a complete complexity dichotomy for deciding whether a graph admits an Eulerian orientation satisfying local Boolean constraints at vertices. We denote this problem by EO. For every fixed finite set of constraints, the problem is either polynomial-time solvable or NP-complete. The classification identifies two new tractable classes. In particular, one such class is obtained by exploiting the promise polymorphism condition used for proving tractability in counting EO [GSS26]. For the hardness proof, we develop a nonconstructive reduction based on unsatisfiable instances and show how failures of the promise conditions yield obstructions to affine and bijunctive tractability, leading to NP-hardness through Schaefer's dichotomy.

Combining our EO dichotomy with the classification of bipartite edge constraint satisfaction problems, we also classify the complexity of general graph orientation problems assuming both unary pinning constraints are available. Every such problem is polynomial-time solvable or NP-complete, except for a residual family of unresolved Δ-matroid parity cases. These results advance the classification of constraint satisfaction problems with restricted variable occurrences and connect their tractability to promise polymorphisms.

AI Disclosure. The main EO dichotomy and its proof were developed and written by the authors. In preparing this version, we used OpenAI Codex to help revise the exposition. Discussions with OpenAI's ChatGPT (GPT-5.5) inspired the always-YES class. Codex also assisted with proof development and checks for extending the EO result to general graph orientations. The authors take full responsibility for the correctness, originality, and integrity of the paper.

1 Introduction

Given an undirected graph G, an *Eulerian orientation* assigns a direction to each edge so that every vertex has equally many incoming and outgoing edges. By the classical Eulerian tour theorem, a graph admits such an orientation if and only if every vertex has even degree [BM08]. Thus deciding whether an Eulerian orientation exists is easy, whereas counting all Eulerian orientations is #P-complete [MW96]. However, restrictions on the local patterns of incoming and outgoing edges can make counting tractable. An example arises from the partition function of the six-vertex model [Pau35, Sla41, Rys63], a widely studied model in statistical physics. Consider a 4-regular graph in which one specified edge at each vertex must be oriented toward that vertex. Counting the Eulerian orientations that satisfy these restrictions is polynomial-time solvable [CFX18]. More generally, local restrictions and weights can be represented by constraint functions at the vertices. Summing the product of these functions over all Eulerian orientations gives the *counting weighted Eulerian orientation* problem, denoted by #EO, introduced in [CFS20].

In a #EO instance, each vertex v of degree d carries a constraint function $f_v : \{0,1\}^d \to \mathbb{C}$, also called a *signature*, evaluated on the local pattern of incoming and outgoing edges. We encode an oriented edge by assigning 0 to its head incidence and 1 to its tail incidence. The *partition function* is $\sum_\sigma \prod_{v \in V} f_v(\sigma|_{E(v)})$, where σ ranges over all Eulerian orientations and $\sigma|_{E(v)}$ is the tuple of incidence values at v. A complexity dichotomy for #EO was first established under a physical symmetry known as *arrow-reversal symmetry* (ARS) [CFS20]: every problem in that setting is either polynomial-time computable or #P-hard. The ARS assumption was later removed in [MWX25], yielding an $\mathrm{FP}^{\mathrm{NP}}$ versus #P-hard classification. The full FP versus #P-hard dichotomy was subsequently obtained in [GSS26].

For a special tractable class of #EO, the counting problem can be reduced to its decision analogue: deciding whether a prescribed local support string extends to a satisfying orientation [MWX25, GSS26]. A particularly interesting 56-ary signature f_{56} provides a concrete example of this tractable class [GSS26]. These results reveal a close connection between counting and feasibility in the EO setting, but leave open the complexity of the general decision problem.

In this paper, we study the *decision problem for constrained Eulerian orientations*, denoted by EO. Fix a set $\mathcal{F}$ of Boolean EO constraints, each of which accepts only strings containing equally many zeros and ones. An instance of EO($\mathcal{F}$) consists of a graph whose vertices are labeled by constraints from $\mathcal{F}$, with specified orders on their incident edges. The task is to determine whether the graph has an orientation whose local configuration is accepted by the constraint at every vertex. The half-weight condition ensures that every such orientation is Eulerian. We establish a complete complexity dichotomy for this problem. The classification identifies two new tractable classes. In particular, we extend the promise polymorphism condition (see Section 2.2) used for tractability of the minority case in #EO [GSS26] to the majority case in decision EO.

Theorem 1.1 (Dichotomy for EO($\mathcal{F}$))**.** *Let $\mathcal{F}$ be a fixed finite set of nonzero EO constraints. If at least one of the following conditions holds:*

1. *every constraint $f \in \mathcal{F}$ is a Δ-matroid;*
2. *every instance of* EO($\mathcal{F}$) *is satisfiable;*
3. *the ternary minority operation* Min_3 *is an upper or lower promise polymorphism of $\mathcal{F}$;*
4. *the ternary majority operation* Maj_3 *is an upper or lower promise polymorphism of $\mathcal{F}$,*

then EO($\mathcal{F}$) *is polynomial-time solvable. Otherwise, it is NP-complete.*

Tractability under condition 1 follows from the tractability of even Δ-matroids in edge constraint satisfaction problems (ECSP) [KKR18], and tractability under condition 3 follows from the

tractability of #EO [GSS26]. The tractable classes in conditions 2 and 4 are new.

The significance of the decision dichotomy for EO is twofold. First, it advances the classification of constraint satisfaction problems (CSPs) with restricted variable occurrences. The study of CSP dichotomies began with Schaefer's Boolean classification [Sch78] and culminated, for finite domains, in the independent dichotomy theorems of Bulatov and Zhuk [Bul17, Zhu20]. These classifications concern CSPs without restrictions on variable occurrences. Requiring each variable to occur exactly twice leads to *edge constraint satisfaction problems* (ECSPs), a setting originating in Feder's work on fanout limitations [Fed01]. An ECSP instance can be represented by a graph in which edges are variables and vertices carry constraints, so each variable participates in the two constraints at its endpoints. The corresponding weighted counting framework is *Holant* [CLX11]. Restricting variable occurrences creates additional tractable cases while limiting the reductions available for proving hardness. Feder's work and the subsequent classification of bipartite ECSP [Fed01, FF06] connect this setting to Δ-matroids. A central question, open for more than two decades, is whether every fixed finite set of Δ-matroid constraints defines a tractable ECSP. Polynomial-time algorithms are known for several special classes, including binary, linear, symmetric, and even Δ-matroids [GIM03, Cor88, KKR18]. The EO problem is a special case of bipartite ECSP, where binary disequality constraints represent the edges of the original graph on one side, while EO constraints represent its vertices on the other side. The disequality constraint allows the two incidence assignments 01 and 10, corresponding to the two orientations of an edge; the EO constraints enforce balance at each vertex. This combination of restricted occurrences and local balance provides a natural setting in which to seek further sources of tractability. Our dichotomy determines the resulting tractability boundary for EO.

The EO dichotomy also provides a basis for studying *graph orientation* (GO) problems, which retain the binary disequality constraints at edges but allow arbitrary Boolean constraints at vertices. For symmetric constraints, which depend only on the number of outgoing edges, a dichotomy with both unary pinning constraints was established in [MAD+26, Theorem 3.12]. Here, the two unary pinning constraints fix an incidence value to 0 or 1. By combining our EO dichotomy with the bipartite ECSP classification, we extend the result to arbitrary Boolean constraints with both pinning constraints, with only the well-known Δ-matroid case remaining unsolved.

Theorem 1.2 (GO trichotomy)**.** *For every fixed finite language of Boolean constraints, graph orientation with both unary pins falls into a polynomial-time branch, an NP-complete branch, or a residual family of unresolved Δ-matroid parity cases. The classification criteria are given by the EO conditions and the bipartite ECSP classification.*

Second, our dichotomy shows how ideas from counting problems and promise CSPs can lead to new algorithms for exact decision problems. Promise polymorphisms, originally called *weak polymorphisms*, were introduced by Austrin, Guruswami, and Håstad [AGH17] and subsequently developed in the algebraic and algorithmic theory of promise constraint satisfaction problems (PCSPs) [BG21, BBKO21, BG19, BGWŽ20]. Unlike an ordinary polymorphism, which preserves a constraint, a promise polymorphism need only map tuples satisfying a stronger constraint to a tuple satisfying a weaker one. This perspective has led to complexity classifications for symmetric Boolean PCSPs [BG21] and #EO [GSS26]. Our new tractable class extends the promise-polymorphism approach used for the minority case of # EO [GSS26] to majority. The relevant promise relaxations are one-sided: an upper relaxation adds only strings of Hamming weight strictly greater than half the arity, while a lower relaxation adds only strings of weight strictly less than half. Accordingly, an upper promise polymorphism may leave the original support only by producing an overweight string, and the lower condition is analogous. The key observation is that the EO incidence structure prevents these local relaxations from introducing new global solutions. Every edge contributes

exactly one 1 across its two incidences, so the total Hamming weight is always $|E|$, equal to the sum of half the vertex degrees. If every constraint is relaxed in the same direction, any local excess or deficit would violate this identity. Every satisfying assignment of the relaxed instance must therefore remain balanced at every vertex and hence satisfy the original constraints. This explains how a promise condition can support exact feasibility algorithms in EO even when the original local supports are not closed under the corresponding operation. It also suggests exploring promise-polymorphism methods together with incidence restrictions in the broader study of ECSPs.

The hardness proof requires different tools from those used in the counting dichotomy. Counting hardness does not imply decision hardness, and reductions based on interpolation or cancellation need not preserve satisfiability. We first recover the two unary pinning constraints through a nonconstructive reduction. For a set of constraints with nonempty supports that is not always satisfiable, we compare a NO instance with a YES instance obtained by rewiring the same vertex constraints. This constructs a gadget from which we obtain either complementary pins or generalized disequalities. In the latter case, complement asymmetry gives pinning constraints. If all constraints are complement-invariant, synchronized pinning pairs simulate the desired pins up to a global bit flip. Thus, the existence of a NO instance supplies the pinning constraints needed for our reductions (Lemma 5.2). With pinnings and generalized disequalities available, our key technical step converts failures of the promise conditions into classical CSP hardness. Failure of both the upper and lower minority conditions yields either NP-hardness directly or a non-affine pairwise opposite constraint (Lemma 5.7). The analogous failure for majority yields either NP-hardness or a non-bijunctive pairwise opposite constraint (Lemma 5.8). If neither step gives hardness directly, the two witnesses together yield an NP-complete CSP by Schaefer's dichotomy.

2 Preliminaries

All constraints have Boolean domain unless stated otherwise. The symbols $\leq_T$ and $\equiv_T$ denote polynomial-time Turing reducibility and equivalence. We write $[P]$ for the indicator of a statement P.

Write $\oplus$ for addition modulo two, $\mathrm{Min}_3(a,b,c) = a \oplus b \oplus c$ for ternary minority, and Maj_3 for ternary majority. Boolean operations act coordinatewise on strings.

2.1 Constraints and Edge CSP

A *constraint* of arity $n \geq 1$ is a function $f : \{0,1\}^n \to \{0,1\}$ with coordinates indexed by $[n] = \{1,\ldots,n\}$. For coordinates labeled $x_1,\ldots,x_n$, we also write $\alpha(x_i) = \alpha_i$.

Definition 2.1 (Support set, $\mathcal{S}(f)$)**.** *The* support *of a constraint f of arity n is $\mathcal{S}(f) = \{\alpha \in \{0,1\}^n \mid f(\alpha) = 1\}$. Its elements are called* support strings.

The *support table* lists the support strings as rows in the specified coordinate order. A *constraint language* is a set of constraints. All constraints are represented by their support tables. In each complexity classification, the language is fixed and finite; the input specifies the graph and the constraint at each vertex.

For a bit string α, write $|\alpha|$ for its length, $\mathrm{wt}(\alpha) = \sum_i \alpha_i$ for its Hamming weight, $\overline{\alpha}$ for its bitwise complement, and $\vec{\alpha}^{\,r}$ for the concatenation of r copies. A constraint has *constant Hamming weight t* if all its support strings have weight t.

The *tensor product* satisfies $(f \otimes g)(\alpha\beta) = f(\alpha)g(\beta)$; for support sets, write $A \otimes B = \{\alpha\beta : \alpha \in A, \beta \in B\}$, where $\alpha\beta$ denotes concatenation.

A constraint is *symmetric* if its value depends only on Hamming weight. For a symmetric constraint of arity n, we write $f = [f_0, f_1, \ldots, f_n]$, where f_i is its value on every string of weight i. The unary *pinning constraints* $\delta_0 = [1, 0]$ and $\delta_1 = [0, 1]$ accept only 0 and 1, respectively.

For $n \geq 1$, the equality constraint $=_n$ and the generalized disequality constraint $\neq_{2n}$ have supports $\mathcal{S}(=_n) = \{\vec{0}^n, \vec{1}^n\}, \mathcal{S}(\neq_{2n}) = \{\vec{0}^n\vec{1}^n, \vec{1}^n\vec{0}^n\}$. In particular, $\neq_2 = [0, 1, 0]$ is binary disequality. We write $\mathcal{EQ} = \{=_n |\ n \geq 1\}$ and $\mathcal{DEQ} = \{\neq_{2n} |\ n \geq 1\}$.

A constraint of arity n is an *EO constraint* if every support string has weight $n/2$. It is *pairwise opposite* if its coordinates admit a fixed pairing such that every support string takes opposite values on each pair. Every pairwise opposite constraint is an EO constraint.

Graphs may have parallel edges and self-loops. Let $E(v)$ be the ordered set of edge incidences at v, and write $\alpha(e)$ for the value at incidence e. We encode an orientation by assigning 1 to the tail incidence and 0 to the head incidence of each edge.

Definition 2.2 (Edge Constraint Satisfaction Problems, ECSP)**.** *An instance of* $\mathrm{ECSP}(\mathcal{F})$ *is a pair* (G, π)*, where* $G = (V, E)$ *and* $\pi(v) = f_v \in \mathcal{F}$ *has arity* $|E(v)|$*. It asks for an assignment* $\phi : E \to \{0, 1\}$ *satisfying* $f_v(\phi(E(v))) = 1$ *for every* v*, where* $\phi(E(v))$ *lists the incident edge values in the local order.*

Every variable occurs exactly twice, once at each incidence. We write $\mathrm{ECSP}(\mathcal{F} \mid \mathcal{G})$ for instances on bipartite graphs $G = (U \mid V, E)$ with constraints from $\mathcal{F}$ on the left-hand side (LHS) U and constraints from $\mathcal{G}$ on the right-hand side (RHS) V.

The *constraint satisfaction problem* is represented by $\mathrm{CSP}(\mathcal{F}) := \mathrm{ECSP}(\mathcal{EQ} \mid \mathcal{F})$.

Definition 2.3 (Constrained Eulerian orientation problems, EO)**.** *For a language* $\mathcal{F}$ *of EO constraints, define* $\mathrm{EO}(\mathcal{F}) := \mathrm{ECSP}(\neq_2 |\ \mathcal{F})$.

2.2 Promise Polymorphisms

We use promise polymorphisms from promise constraint satisfaction problems (PCSPs) [BG21, BBKO21] to express the minority and majority conditions.

A *Boolean promise constraint* is a pair (f, g) of constraints of the same arity with $\mathcal{S}(f) \subseteq \mathcal{S}(g)$. A k-ary Boolean operation p is a *polymorphism* of f if $p(\alpha^{(1)}, \ldots, \alpha^{(k)}) \in \mathcal{S}(f)$ whenever $\alpha^{(1)}, \ldots, \alpha^{(k)} \in \mathcal{S}(f)$, and a *promise polymorphism* of (f, g) if these outputs belong to $\mathcal{S}(g)$. For a set $\mathcal{P}$ of promise constraints, write $p \in \mathsf{Pol}(\mathcal{P})$ when p is a promise polymorphism of every member of $\mathcal{P}$.

Let f have arity n and constant Hamming weight t. Its *upper relaxation* $f^{\uparrow}$ and *lower relaxation* $f^{\downarrow}$ are defined by

$$\mathcal{S}(f^{\uparrow}) = \mathcal{S}(f) \cup \{\alpha \in \{0,1\}^n : \mathrm{wt}(\alpha) > t\},$$
$$\mathcal{S}(f^{\downarrow}) = \mathcal{S}(f) \cup \{\alpha \in \{0,1\}^n : \mathrm{wt}(\alpha) < t\}.$$

For a language $\mathcal{F}$ of constant-weight constraints and $s \in \{\uparrow, \downarrow\}$, set

$$\mathcal{P}^s(\mathcal{F}) = \{(f, f^s) : f \in \mathcal{F}\}.$$

We call p an *upper promise polymorphism* of $\mathcal{F}$ if $p \in \mathsf{Pol}(\mathcal{P}^{\uparrow}(\mathcal{F}))$, and a *lower promise polymorphism* if $p \in \mathsf{Pol}(\mathcal{P}^{\downarrow}(\mathcal{F}))$. For a single constraint f, take $\mathcal{F} = \{f\}$.

2.3 Gadgets and Reductions

An $\mathcal{F} \mid \mathcal{G}$-*gate* is an ECSP($\mathcal{F} \mid \mathcal{G}$) instance with ordered dangling edges. It realizes the constraint accepting exactly the assignments to those edges that extend to a satisfying assignment. The realization is on the LHS or RHS if all dangling edges are incident to that side. Unless stated otherwise, gates have size independent of the input instance; their realized constraints can be added on the corresponding side without changing the complexity. We abbreviate $\mathcal{F} \cup \{f\}$ by $\mathcal{F}, f$, and similarly for unions of languages. The proofs of Lemmas 2.4 to 2.6 are given in Appendix A.

Pinning. Pinning a coordinate to $t \in \{0, 1\}$ fixes its value to t and removes it. The remaining coordinates retain their labels and order; a fully pinned constraint is evaluated directly.

Deduplication. For a fixed $l \geq 1$, an *l-multiple* f^l of an n-ary constraint f is defined, up to a coordinate permutation, by

$$\mathcal{S}(f^l) = \{\vec{\alpha_1}^{\,l} \cdots \vec{\alpha_n}^{\,l} : \alpha \in \mathcal{S}(f)\}.$$

We call f a *deduplication* of f^l.

Lemma 2.4. *For any languages $\mathcal{F}, \mathcal{G}$ and any l-multiple f^l of f,*

$$\mathrm{ECSP}(\mathcal{F} \mid \mathcal{G}, f) \leq_T \mathrm{ECSP}(\mathcal{F} \mid \mathcal{G}, f^l).$$

In particular, $=_l$ is an l-multiple of $=_1$.

Disequalities.

Lemma 2.5. *For distinct coordinates i, j of an n-ary constraint f, the n-ary constraint $f_{i\neq j}(\alpha) = f(\alpha)[\alpha_i \neq \alpha_j]$ is realizable on the RHS of* $\mathrm{ECSP}(\neq_2|\neq_4, f)$.

Lemma 2.6. *Let $\mathcal{F}$ be a finite set of constraints. For every fixed $n \geq 2$,*

$$\mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}) \leq_T \mathrm{ECSP}(\neq_2| \mathcal{F}, \neq_{2n}).$$

Lemma 2.7 ([CFS20, proof of Lemma 5.8])**.** *Let $\mathcal{F}$ be a finite set of pairwise opposite constraints. Then*

$$\mathrm{CSP}(\mathcal{F}) \leq_T \mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{F}).$$

2.4 Known Results

A constraint f of arity r is *0-valid* if $f(\vec{0}^r) = 1$, and *1-valid* if $f(\vec{1}^r) = 1$. A constraint is *Horn* if its support is closed under coordinatewise AND, and *dual-Horn* if its support is closed under coordinatewise OR. It is *affine* if it admits Min_3 as a polymorphism, and *bijunctive* if it admits Maj_3 as a polymorphism.

Write $\alpha\Delta\beta = \{i : \alpha_i \neq \beta_i\}$ and let $\alpha \oplus S$ mean flipping the coordinates in a set S. A constraint f is a Δ-*matroid* if its support is nonempty and, for all $\alpha, \beta \in \mathcal{S}(f)$ and $i \in \alpha\Delta\beta$, there exists $j \in \alpha\Delta\beta$ (possibly $j = i$) such that $\alpha \oplus \{i, j\} \in \mathcal{S}(f)$. It is an *even* Δ-*matroid* if, in addition, all its support strings have Hamming weights of the same parity.

Theorem 2.8 (Schaefer's Boolean CSP dichotomy [Sch78])**.** *Let $\mathcal{F}$ be a finite set of nonzero Boolean constraints. If $\mathcal{F}$ is contained in one of the six classes of* 0*-valid,* 1*-valid, Horn, dual-Horn, affine, or bijunctive constraints, then* $\mathrm{CSP}(\mathcal{F})$ *is solvable in polynomial time. Otherwise, it is NP-complete.*

Lemma 2.9 ([KKR18, Theorem 8])**.** *An ECSP instance whose constraints are all even Δ-matroids can be decided in polynomial time.*

3 Examples and Proof Sketch

Examples of new tractable classes. Conditions 2 and 4 of Theorem 1.1 define two new tractable classes: always-YES languages and languages admitting Maj_3 as an upper or lower promise polymorphism. For condition 2, let f have support

$$\mathcal{S}(f) = \{01, 10\}^{\otimes 3} \cup \{110100\}.$$

Then every instance of $\mathrm{EO}(\{f\})$ is satisfiable, but f is not a Δ-matroid and admits neither Min_3 nor Maj_3 as an upper or lower promise polymorphism.

For condition 4, let g have support

$$\mathcal{S}(g) = \{111000, 100110, 100101\} \otimes \{0101, 0110, 1001\}.$$

Here Maj_3 is a lower promise polymorphism of g, and $\{g\}$ satisfies none of conditions 1–3. Moreover, g is not bijunctive, so the lower relaxation is essential.

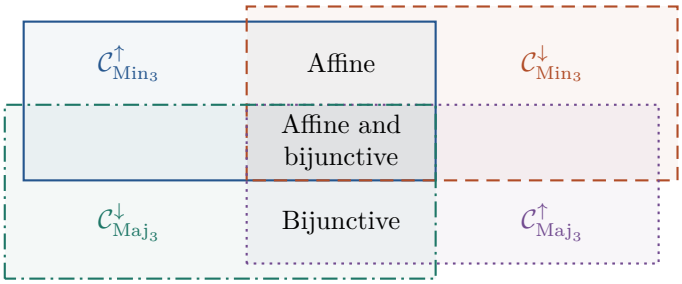


Figure 1: The four classes of EO constraints $\mathcal{C}^s_p = \{f : p \in \mathsf{Pol}(\{(f, f^s)\})\}$, where $p \in \{\mathrm{Min}_3, \mathrm{Maj}_3\}$ and $s \in \{\uparrow, \downarrow\}$.

3.1 Proof Sketch and Technical Highlights

The tractable direction has three ingredients. Every EO Δ-matroid is even, as is $\neq_2$, so condition 1 follows from the KKR algorithm in Lemma 2.9. Condition 2 admits the constant YES algorithm. In Section 4, for conditions 3 and 4, an LP restricts the local supports to affine or bijunctive constraints while preserving all solutions; Gaussian elimination or 2-SAT then decides the instance.

For the hardness direction, fix a finite language $\mathcal{F}$ of nonzero EO constraints, and write $\mathrm{EO}^c(\mathcal{F})$ for the problem with both unary pinnings. If condition 2 fails, a NO instance lets us simulate both pinnings. A transition from a YES pairing to a fixed NO pairing produces either a complementary pinning pair or $\neq_4$. In both cases, we obtain $\mathrm{EO}^c(\mathcal{F}) \leq_T \mathrm{EO}(\mathcal{F})$ (Lemma 5.2).

We then prove the dichotomy for $\mathrm{EO}^c(\mathcal{F})$ (Theorem 5.3). Pinnings let us fix coordinates when constructing gadgets. Failure of the Δ-matroid condition realizes $\neq_4$, and hence all generalized disequalities (Lemma 5.4). If the minority and majority tractability conditions also fail, Lemmas 5.7 and 5.8 yield NP-hardness, either directly or through a reduction from an NP-complete CSP. Combining this result with the first reduction proves NP-completeness of $\mathrm{EO}(\mathcal{F})$ whenever all four tractability conditions fail.

From counting EO to a bijunctive decision class. Our LP algorithm extends the method of Guan, Shao, and Shi [GSS26] for #EO. In our PCSP formulation, their minority method gives affine LP-restricted supports. For decision problems, bijunctive supports also suffice because 2-SAT decides feasibility. We show that the same LP restriction preserves all solutions and yields bijunctive constraints when Maj_3 is an upper or lower promise polymorphism of the language (Theorem 4.5).

Obtaining pinnings from a YES–NO transition. A NO instance contains enough information to recover the power of pinnings. Fix its vertex constraints and local port orders. Nonempty supports let us choose one accepted string at each vertex; half weight then lets us pair every 0 port with a 1 port, producing a YES instance on the same vertices. Any two pairings are connected by switches, each changing two pairs on four ports. Along a sequence from the YES pairing to the NO pairing, consider the first switch that destroys satisfiability.

Cutting these two edges leaves a nonempty four-port constraint g. Global balance forces weight two, and the NO pairing excludes four of the six weight-two strings. After ordering the ports,

$$\varnothing \neq \mathcal{S}(g) \subseteq \{0011, 1100\}.$$

A singleton support gives a complementary pair of pins by contracting one 0 port with one 1 port. Otherwise $g = \neq_4$, which supplies all generalized disequalities. A constraint that is not invariant under complementation then selects one of the two pinning patterns. If every constraint is complement-invariant, generalized disequality instead synchronizes all pinning pairs; a global bit flip recovers their prescribed values. This proves pin elimination (Lemma 5.2).

Failed PCSP conditions yield hardness. Assume that pinnings and $\mathcal{DEQ}$ are available. If Min_3 is neither an upper nor a lower promise polymorphism of $\mathcal{F}$, Lemma 5.7 gives either NP-hardness directly or a pairwise opposite constraint that is not affine. If Maj_3 is neither an upper nor a lower promise polymorphism of $\mathcal{F}$, Lemma 5.8 gives either NP-hardness directly or a pairwise opposite constraint that is not bijunctive. If neither lemma gives hardness directly, the resulting constraints yield an NP-complete CSP by Schaefer's dichotomy.

4 An LP-Based Framework for the Minority and Majority Cases

We use the LP and minority support-restriction argument of Guan, Shao, and Shi [GSS26], and prove its majority analogue. Under the hypothesis below, the LP restricts local supports to affine or bijunctive constraints while preserving all satisfying orientations. Gaussian elimination or 2-SAT then decides feasibility.

Proposition 4.1. *Let $\mathcal{F}$ be a finite set of EO constraints. If some $p \in \{\mathrm{Min}_3, \mathrm{Maj}_3\}$ is an upper or lower promise polymorphism of $\mathcal{F}$, then* $\mathrm{EO}(\mathcal{F})$ *is polynomial-time solvable.*

4.1 The Linear System

Write an instance of $\mathrm{EO}(\mathcal{F})$ in its orientation form $\Omega = (G, \pi)$, where $G = (V, E)$, and let $f_v = \pi(v)$. For each edge $e = (u, v)$, denote its incidences by $e_1 \in E(u)$ and $e_2 \in E(v)$. These incidences are distinct even when e is a self-loop. Thus $\alpha(e_1)$ and $\beta(e_2)$ refer to the corresponding coordinates of support strings at its endpoints.

The variable $x_{v,\alpha}$ represents the mass assigned to local string α at v. Vertex equations make these masses a probability distribution; edge equations require the two incidence marginals of an edge to sum to one. Consider the following system.

$$
\begin{aligned}
&\textbf{Variables:} && x_{v,\alpha} \geq 0 && \text{for every } v \in V \text{ and } \alpha \in \mathcal{S}(f_v),\\
&\textbf{Vertex equations:} && \sum_{\alpha \in \mathcal{S}(f_v)} x_{v,\alpha} = 1 && \text{for every } v \in V,\\
&\textbf{Edge equations:} && \sum_{\alpha \in \mathcal{S}(f_u)} \alpha(e_1) x_{u,\alpha} + \sum_{\beta \in \mathcal{S}(f_v)} \beta(e_2) x_{v,\beta} = 1 && \text{for every } e = (u,v) \in E.
\end{aligned}
\tag{1}
$$

Write $\boldsymbol{x} = (x_{v,\alpha})_{v \in V, \alpha \in \mathcal{S}(f_v)}$, and let $\mathsf{P}(\Omega)$ denote the feasible region of (1). The vertex equations and nonnegativity imply that every variable lies in $[0,1]$, so $\mathsf{P}(\Omega)$ is a polytope.

Definition 4.2 (LP-feasible support, $\mathcal{L}(f_v)$)**.** *A string $\alpha \in \mathcal{S}(f_v)$ is* LP-feasible *if $x_{v,\alpha} > 0$ for some $\boldsymbol{x} \in \mathsf{P}(\Omega)$. Let $\mathcal{L}(f_v)$ be the set of these strings; it is empty when $\mathsf{P}(\Omega)$ is empty.*

The support $\mathcal{L}(f_v)$ depends on the vertex and the instance. If $\mathsf{P}(\Omega)$ is nonempty, the vertex equations ensure that each $\mathcal{L}(f_v)$ is nonempty. For a constraint f of arity r and a set $\mathcal{T} \subseteq \{0,1\}^r$, write $f|_{\mathcal{T}}$ for the constraint with support $\mathcal{S}(f) \cap \mathcal{T}$.

Lemma 4.3. *If $\mathsf{P}(\Omega)$ is empty, then Ω is unsatisfiable. Otherwise, replacing each f_v by $f_v|_{\mathcal{L}(f_v)}$ yields an equisatisfiable instance Ω' that can be constructed in polynomial time.*

Proof. Every satisfying orientation induces a feasible indicator vector, so its local strings belong to $\mathcal{L}(f_v)$. Together with $\mathcal{L}(f_v) \subseteq \mathcal{S}(f_v)$, this proves both satisfiability claims. Compute each $\mathcal{L}(f_v)$ by maximizing $x_{v,\alpha}$ over $\mathsf{P}(\Omega)$ and retaining strings with positive optimum. This requires polynomially many linear programs of polynomial size, solvable exactly in polynomial time [Kar84]. □

4.2 Properties of the LP-Feasible Support

We first show that every triple receiving positive mass under a feasible vector has balanced minority and majority outputs. We then apply the relevant promise polymorphism to the local distributions to prove closure of the LP-feasible supports.

We encode a bit $a \in \{0,1\}$ by its sign $\widehat{a} = 2a - 1$ and apply this notation coordinatewise to strings. For $\alpha \in \{0,1\}^r$, define

$$\Delta(\alpha) = \sum_{i=1}^{r} \widehat{\alpha}_i = 2\,\mathrm{wt}(\alpha) - r.$$

Thus Δ records the signed departure from half the arity. For Boolean inputs a, b, c, the signed forms of minority and majority are

$$\widehat{\mathrm{Min}_3(a,b,c)} = \widehat{a}\widehat{b}\widehat{c}, \qquad \widehat{\mathrm{Maj}_3(a,b,c)} = \tfrac{1}{2}(\widehat{a} + \widehat{b} + \widehat{c} - \widehat{a}\widehat{b}\widehat{c}).$$

Consequently,

$$\Delta(\mathrm{Min}_3(\alpha,\beta,\gamma)) = \sum_{i=1}^{r} \widehat{\alpha}_i \widehat{\beta}_i \widehat{\gamma}_i,$$

$$2\Delta(\mathrm{Maj}_3(\alpha,\beta,\gamma)) + \Delta(\mathrm{Min}_3(\alpha,\beta,\gamma)) = \Delta(\alpha) + \Delta(\beta) + \Delta(\gamma).$$

For three EO support strings, the right-hand side of the second identity is zero.

Lemma 4.4. *Let $\Omega = (G, \pi)$ be an instance of $\mathrm{EO}(\mathcal{F})$, where $\pi(v) = f_v$. Suppose that $\mathcal{F}$ satisfies the hypothesis of Proposition 4.1. Then, for every $\boldsymbol{x} \in \mathsf{P}(\Omega)$, every $v \in V$, and all $\alpha, \beta, \gamma \in \mathcal{S}(f_v)$,*

$$x_{v,\alpha} x_{v,\beta} x_{v,\gamma} \Delta(\mathrm{Min}_3(\alpha, \beta, \gamma)) = 0,$$
$$x_{v,\alpha} x_{v,\beta} x_{v,\gamma} \Delta(\mathrm{Maj}_3(\alpha, \beta, \gamma)) = 0.$$

Proof. For every incidence $i \in E(v)$, define its signed marginal

$$m_{v,i} = \sum_{\alpha \in \mathcal{S}(f_v)} \widehat{\alpha}(i) x_{v,\alpha}.$$

The vertex and edge equations imply $m_{u,e_1} + m_{v,e_2} = 0$ for the two incidences of every edge $e = (u, v)$. Cubing preserves this cancellation. Summing over edges and then grouping incidences by vertex gives

$$\sum_{v \in V} \sum_{i \in E(v)} m_{v,i}^3 = 0. \tag{2}$$

Expanding at each vertex gives

$$\begin{aligned}
\sum_{i \in E(v)} m_{v,i}^3 &= \sum_{\alpha,\beta,\gamma \in \mathcal{S}(f_v)} x_{v,\alpha} x_{v,\beta} x_{v,\gamma} \sum_{i \in E(v)} \widehat{\alpha}(i) \widehat{\beta}(i) \widehat{\gamma}(i) \\
&= \sum_{\alpha,\beta,\gamma \in \mathcal{S}(f_v)} x_{v,\alpha} x_{v,\beta} x_{v,\gamma} \Delta(\mathrm{Min}_3(\alpha, \beta, \gamma)) \\
&= -2 \sum_{\alpha,\beta,\gamma \in \mathcal{S}(f_v)} x_{v,\alpha} x_{v,\beta} x_{v,\gamma} \Delta(\mathrm{Maj}_3(\alpha, \beta, \gamma)).
\end{aligned}$$

The last equality uses the fact that all input strings have half weight.

If Min_3 is an upper promise polymorphism of $\mathcal{F}$, every minority output has nonnegative imbalance. All terms in the middle sums are therefore nonnegative. Their global sum is zero by (2), so every term must vanish. If Min_3 is a lower promise polymorphism of $\mathcal{F}$, these terms are nonpositive, with the same conclusion. The identity relating the two imbalances then also makes every majority term vanish.

If Maj_3 is an upper or lower promise polymorphism of $\mathcal{F}$, apply the same argument to the last sum: its terms have a common sign and their global sum is zero. Both asserted products therefore vanish in all four cases. □

Theorem 4.5. *Let $\Omega = (G, \pi)$ be an instance of $\mathrm{EO}(\mathcal{F})$ with $\pi(v) = f_v$, and suppose that $\mathsf{P}(\Omega)$ is nonempty. Then the following statements hold at every vertex v.*

1. *If Min_3 is an upper or lower promise polymorphism of $\mathcal{F}$, then $f_v|_{\mathcal{L}(f_v)}$ is affine.*
2. *If Maj_3 is an upper or lower promise polymorphism of $\mathcal{F}$, then $f_v|_{\mathcal{L}(f_v)}$ is bijunctive.*

Proof. Let p be the minority or majority operation specified by the hypothesis, and fix a vertex v and strings $\alpha, \beta, \gamma \in \mathcal{L}(f_v)$. We prove that $p(\alpha, \beta, \gamma) \in \mathcal{L}(f_v)$.

A feasible vector supporting all three strings. Choose feasible vectors $\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, \boldsymbol{x}^{(3)}$ with $x^{(1)}_{v,\alpha}, x^{(2)}_{v,\beta}, x^{(3)}_{v,\gamma} > 0$. Their average

$$\boldsymbol{x}^* = \tfrac{1}{3}(\boldsymbol{x}^{(1)} + \boldsymbol{x}^{(2)} + \boldsymbol{x}^{(3)}) \in \mathsf{P}(\Omega)$$

gives positive mass to all three strings at v. Neither the strings nor their witnessing vectors need be distinct.

Applying the operation to local distributions. At every vertex w, independently sample three strings from the distribution $x^*_{w,\cdot}$ and apply p coordinatewise. This is a transformation of local distributions; it does not assume a distribution on satisfying orientations. Define the resulting masses by

$$\mu_{w,\zeta} = \sum_{\substack{a,b,c \in \mathcal{S}(f_w) \\ p(a,b,c)=\zeta}} x^*_{w,a} x^*_{w,b} x^*_{w,c}, \qquad \zeta \in \mathcal{S}(f_w).$$

These masses are nonnegative. Since p is an upper or lower promise polymorphism of $\mathcal{F}$, an output outside $\mathcal{S}(f_w)$ has strictly positive or strictly negative imbalance, respectively. By Lemma 4.4, the triple has mass $x^*_{w,a} x^*_{w,b} x^*_{w,c} = 0$. No positive mass is therefore lost by indexing $\boldsymbol{\mu}$ only by support strings. In particular,

$$\sum_{\zeta \in \mathcal{S}(f_w)} \mu_{w,\zeta} = \sum_{a,b,c \in \mathcal{S}(f_w)} x^*_{w,a} x^*_{w,b} x^*_{w,c} = \left(\sum_{a \in \mathcal{S}(f_w)} x^*_{w,a} \right)^3 = 1.$$

Thus all vertex equations hold.

Preserving the edge equations. For an incidence $i \in E(w)$, let

$$m_{w,i} = \sum_{a \in \mathcal{S}(f_w)} \widehat{a}(i) x^*_{w,a}$$

be its signed marginal. The bit identities preceding Lemma 4.4 give

$$\sum_{\zeta \in \mathcal{S}(f_w)} \widehat{\zeta}(i) \mu_{w,\zeta} = \begin{cases} m^3_{w,i}, & p = \mathrm{Min}_3, \\ (3m_{w,i} - m^3_{w,i})/2, & p = \mathrm{Maj}_3 . \end{cases}$$

Indeed, the expectation of the product of the three signed bits is $m^3_{w,i}$; the majority identity also contributes their three expectations, each equal to $m_{w,i}$. The zero-mass property above permits taking these expectations over all triples.

Both displayed functions of $m_{w,i}$ are odd. For an edge $e = (u, w)$ with incidences e_1, e_2, feasibility of $\boldsymbol{x}^*$ gives $m_{u,e_1} + m_{w,e_2} = 0$. The two signed marginals of $\boldsymbol{\mu}$ therefore also sum to zero. Together with the vertex equations, this is precisely the edge equation for $\boldsymbol{\mu}$. The same argument applies when $u = w$, because the incidences remain distinct. Hence $\boldsymbol{\mu} \in \mathsf{P}(\Omega)$.

Finally, put $\tau = p(\alpha, \beta, \gamma)$. This triple has positive mass, so the zero-mass property implies $\tau \in \mathcal{S}(f_v)$. Moreover,

$$\mu_{v,\tau} \geq x^*_{v,\alpha} x^*_{v,\beta} x^*_{v,\gamma} > 0.$$

Thus $\tau \in \mathcal{L}(f_v)$, proving the required closure. Closure under minority characterizes affine constraints, and closure under majority characterizes bijunctive constraints. □

We now give the algorithm establishing Proposition 4.1.

Proof of Proposition 4.1. By Lemma 4.3, Algorithm 1 either detects an infeasible LP and rejects, or constructs an equisatisfiable restricted instance Ω'. By Theorem 4.5, its local constraints are affine in the minority case and bijunctive in the majority case. Binary disequality belongs to both classes.

An affine support is encoded by translating a support string to zero, finding linear equations for the resulting subspace over the two-element field $\mathbb{F}_2$, and translating the equations back. A bijunctive support is encoded by the conjunction of all unary and binary clauses satisfied by its strings. Since arities are bounded, these encodings can be constructed in polynomial time. Gaussian elimination or 2-SAT then solves Ω' in polynomial time. □

Algorithm 1 LP-based algorithm

INPUT: An instance $\Omega = (G, \pi)$ of $\mathrm{EO}(\mathcal{F})$, where $\mathcal{F}$ satisfies the hypothesis of Proposition 4.1
OUTPUT: Whether Ω has a satisfying Eulerian orientation

Test feasibility of (1).
if $\mathsf{P}(\Omega) = \emptyset$ **then**
    return No
end if
for each vertex $v \in V(G)$ **do**
    for each string $\alpha \in \mathcal{S}(f_v)$ **do**
        Solve the linear program $\max_{\boldsymbol{x} \in \mathsf{P}(\Omega)} x_{v,\alpha}$
    end for
    Set $\mathcal{L}(f_v)$ to the strings whose optimum is positive
end for
Construct $\Omega' = (G, \pi')$, where $\pi'(v) = f_v|_{\mathcal{L}(f_v)}$.
Solve Ω' by Gaussian elimination (minority case) or 2-SAT (majority case).
return the resulting satisfiability decision.

5 NP-completeness of $\mathrm{EO}(\mathcal{F})$

Throughout this section, the original language $\mathcal{F}$ is fixed and finite and consists of nonzero constraints; the auxiliary family $\mathcal{DEQ}$ may have unbounded arity. All problems considered belong to NP, and the reductions are satisfiability-preserving constructions of polynomial size.

For a language $\mathcal{F}$ of EO constraints, we write

$$\mathrm{EO}^c(\mathcal{F}) := \mathrm{ECSP}(\neq_2 |\ \mathcal{F}, \delta_0, \delta_1). \tag{3}$$

5.1 Pinnings from a NO Instance

Lemma 5.1. *Let $\mathcal{F}$ be a finite set of EO constraints. Then $\mathrm{EO}^c(\mathcal{F}) \equiv_T \mathrm{EO}(\mathcal{F}, \delta_0 \otimes \delta_1)$.*

Proof. In the bipartite representation, LHS disequalities force exactly half the edges to have value 1. Since each RHS EO constraint requires half its incident values to be 1, a satisfying assignment requires equally many δ_0 and δ_1 vertices. If their numbers differ, output two $\delta_0 \otimes \delta_1$ vertices joined at corresponding ports through separate LHS copies of $\neq_2$, an unsatisfiable target instance. Otherwise, pair the unary vertices and merge each pair into $\delta_0 \otimes \delta_1$. Conversely, split each such pair into its unary factors. Both constructions preserve satisfiability. □

Lemma 5.2 (Simulating pinnings)**.** *Let $\mathcal{F}$ be a fixed finite set of nonzero EO constraints. If $\mathrm{EO}(\mathcal{F})$ has an unsatisfiable instance, then*

$$\mathrm{EO}^c(\mathcal{F}) \leq_T \mathrm{EO}(\mathcal{F}).$$

Proof. By Lemma 5.1, it suffices to show that

$$\mathrm{ECSP}(\neq_2 |\ \mathcal{F}, \delta_0 \otimes \delta_1) \leq_T \mathrm{ECSP}(\neq_2 |\ \mathcal{F}).$$

All gadgets below are $\neq_2 |\ \mathcal{F}$-gates realizing constraints on the RHS, unless additional RHS constraints are explicitly specified.

A switch from a satisfiable pairing to an unsatisfiable one. Since $\mathcal{F}$ is not universally satisfiable, fix an unsatisfiable instance Ω of ECSP($\neq_2|\ \mathcal{F}$). Keep its RHS vertices, their constraints f_v, and their local input orders fixed. Each LHS vertex pairs two RHS ports, so the LHS vertices specify a perfect matching on these ports. The instance Ω depends only on the fixed language $\mathcal{F}$, so the gadget obtained from it has constant size.

We first construct a satisfiable instance on the same RHS vertices. Choose $\alpha_v \in \mathcal{S}(f_v)$ at each vertex, using the assumption that input constraints are nonzero. Because every f_v is an EO constraint, the chosen strings assign 0 and 1 to equally many ports. Pair each 0 port with a 1 port through an LHS copy of $\neq_2$. The resulting instance $\Omega^{(0)}$ is satisfied by the chosen strings.

Any two perfect matchings on the ports are connected by switches, each replacing two pairs by another pairing of the same four ports. Indeed, if a target pair $\{a, b\}$ is absent and the current pairs containing its ports are $\{a, c\}$ and $\{b, d\}$, replace them by $\{a, b\}$ and $\{c, d\}$. This increases the number of pairs shared with the target matching. Thus a sequence of switches transforms $\Omega^{(0)}$ into Ω.

At the first loss of satisfiability, two consecutive instances Ω^+ and Ω^- are satisfiable and unsatisfiable, respectively. Label the four affected ports p_1, p_2, p_3, p_4 so that their pairs are $\{p_1, p_2\}, \{p_3, p_4\}$ in Ω^- and $\{p_1, p_4\}, \{p_2, p_3\}$ in Ω^+.

The four-port gate. Remove the two switched LHS vertices, leaving four dangling edges in the order p_1, p_2, p_3, p_4. The resulting gate realizes a constraint g with nonempty support, since Ω^+ is satisfiable. If Ω has m LHS vertices, balance at the RHS vertices forces a total of m ones on their ports. The remaining $m-2$ LHS disequalities account for $m-2$ of those ones. Every support string of g therefore has weight 2.

No $\alpha \in \mathcal{S}(g)$ satisfies both $\alpha_1 \neq \alpha_2$ and $\alpha_3 \neq \alpha_4$: such a string would extend to a satisfying assignment of Ω^-. Of the six weight-two strings, only 0011 and 1100 avoid these two disequalities. Hence

$$\varnothing \neq \mathcal{S}(g) \subseteq \{0011, 1100\}.$$

Producing a pinning pair. If $\mathcal{S}(g)$ is a singleton, apply disequality contraction to two coordinates assigned 0 and 1, respectively, by its unique support string. Up to a permutation of the two remaining coordinates, the resulting constraint is $\delta_0 \otimes \delta_1$, giving the required reduction by gadget replacement.

Otherwise, $g = \neq_4$. By Lemma 2.6, every $\neq_{2k} \in \mathcal{DEQ}$ is now realizable on the RHS by a gadget constructible in polynomial time. Suppose first that some $f \in \mathcal{F}$ is not invariant under bitwise complementation. Choose $\alpha \in \mathcal{S}(f)$ with $\overline{\alpha} \notin \mathcal{S}(f)$, and let f have arity $2n$.

Take one RHS copy each of f and $\neq_{2n+2}$. Connect the n ports of f at which α is 1 to n ports in the first equality block of $\neq_{2n+2}$, and its other n ports to n ports in the second block. Use a separate LHS copy of $\neq_2$ for each connection. Leave one port of each block dangling, in that order. The external assignment 01 induces α at f, whereas 10 induces $\overline{\alpha}$. Thus this gate realizes $\delta_0 \otimes \delta_1$.

Removing pinnings under complement symmetry. Finally, suppose that every $f \in \mathcal{F}$ is invariant under bitwise complementation. Consider an instance of ECSP($\neq_2|\ \mathcal{F}, \delta_0 \otimes \delta_1$) with $k \geq 1$ occurrences of $\delta_0 \otimes \delta_1$; if $k = 0$, no change is needed. Synchronize these pairs by replacing them with one copy of $\neq_{2k}$: attach their former first ports to its first block and their second ports to its second block.

Every satisfying assignment of the original instance satisfies the new one. Conversely, a solution of the new instance gives either 01 at every former pair or 10 at every pair. In the latter case, complement all edge values. This preserves $\mathcal{F}$ and $\neq_2$ and restores the required 01 values. Thus

the instances are equisatisfiable. Substituting the gate for $\neq_{2k}$ completes the polynomial-time reduction. □

5.2 Dichotomy for EO^c

Theorem 5.3. *Let $\mathcal{F}$ be a fixed finite set of nonzero EO constraints. If every $f \in \mathcal{F}$ is a Δ-matroid, or some $p \in \{\mathrm{Min}_3, \mathrm{Maj}_3\}$ is an upper or lower promise polymorphism of $\mathcal{F}$, then $\mathrm{EO}^c(\mathcal{F})$ is polynomial-time solvable. Otherwise, it is NP-complete.*

The hardness proof first uses a failed exchange to obtain $\neq_4$, and then uses the failures for Min_3 and Maj_3 to obtain CSP obstructions.

Lemma 5.4. *Let $\mathcal{F}$ be a finite set of nonzero constraints such that every $f \in \mathcal{F}$ has constant Hamming weight. If $\mathcal{F}$ contains a constraint that is not a Δ-matroid, then $\neq_4$ is realizable on the RHS by a $\neq_2|\ \mathcal{F}, \delta_0, \delta_1$-gate. Consequently,*

$$\mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1) \leq_T \mathrm{ECSP}(\neq_2|\ \mathcal{F}, \delta_0, \delta_1).$$

Proof. Choose a constant-weight constraint $f \in \mathcal{F}$ that is not a Δ-matroid. Call an ordered pair $(\alpha, \beta) \in \mathcal{S}(f)^2$ *bad* if some $i_0 \in \alpha \Delta \beta$ satisfies

$$\alpha \oplus \{i_0, j\} \notin \mathcal{S}(f) \qquad \text{for every } j \in \alpha\Delta\beta,\ j \neq i_0.$$

We call i_0 a bad coordinate. A one-coordinate flip cannot preserve Hamming weight, so failure of the Δ-matroid axiom gives a bad pair. Its Hamming distance is at least four. Choose a bad pair of minimum distance, with bad coordinate i_0.

Pin all coordinates on which α and β agree, and let f' be the resulting constraint. Retain the original indices of its coordinates. The restricted strings α' and β' are complementary, and f' still has constant weight. Thus, if m is its arity, both strings have weight $m/2$, and $m \geq 4$. Every support string of f' extends uniquely, using the pinned values, to a support string of f.

If $\mathcal{S}(f') = \{\alpha', \beta'\}$, a coordinate permutation gives $f' = \neq_m$. Apply $m/2 - 2$ disequality contractions, each joining one port from each equality block, to obtain $\neq_4$. Otherwise, consider any $\gamma' \in \mathcal{S}(f') \setminus \{\alpha', \beta'\}$, and let $\gamma \in \mathcal{S}(f)$ be its extension by the pinned values. Minimality of the bad pair imposes the following restrictions.

Only β' differs from α' at the bad coordinate. We have $\gamma'_{i_0} = \alpha'_{i_0}$. Indeed, if $\gamma'_{i_0} = \beta'_{i_0}$, then $\alpha\Delta\gamma$ is a proper subset of $\alpha\Delta\beta$ containing i_0. No exchange at i_0 within this smaller set is possible, because none was possible within the larger set. Hence (α, γ) is a smaller bad pair, a contradiction.

Every additional string is at distance two from β'. Suppose instead that $\mathrm{wt}(\gamma' \oplus \beta') \geq 4$. If an exchange at i_0 took γ' to another support string, that string would have value β'_{i_0} at i_0. By the preceding paragraph it must be β', which is impossible at distance at least four. Extending by the pinned values therefore gives a bad pair (γ, β) smaller than (α, β). This is again a contradiction.

The remaining constraint has arity four. If $m \geq 6$, complementation and the preceding paragraph give $\mathrm{wt}(\alpha' \oplus \gamma') = m - 2 \geq 4$. Fix any coordinate $k \in \alpha'\Delta\gamma'$. A two-coordinate exchange from α' at k within this difference set would produce a support string θ' at distance two from α'. It is neither α' nor β', but its distance from β' is $m - 2 \geq 4$, contradicting the preceding paragraph. Thus (α, γ) would be a smaller bad pair. We conclude that $m = 4$.

Suppose first that $\alpha'_{i_0} = 0$. Permute the coordinates so that i_0 is first, $\alpha' = 0011$, and $\beta' = 1100$. The additional support strings have first coordinate zero and weight two, so the support is one of

$$\{0011, 0101, 1100\}, \qquad \{0011, 0110, 1100\}, \qquad \{0011, 0101, 0110, 1100\}.$$

The first two differ only by exchanging coordinates three and four. Write h_3 and h_4 for the first and third constraints, respectively.

Take two copies of h_4 on the RHS. Join their third ports through one LHS copy of $\neq_2$, and their fourth ports through another. Order the dangling ports as the first two ports of the first copy followed by those of the second. The resulting support is

$$\{0011, 0101, 1100\} = \mathcal{S}(h_3).$$

To see this, a prefix 00 has suffix 11 and must pair with prefix 11, whose suffix is 00. The only other possibility pairs the two suffixes 01 and 10, both of which have prefix 01. Applying the same construction to h_3 removes this last possibility, because h_3 has suffix 01 but not suffix 10. Its resulting support is therefore $\{0011, 1100\}$, giving $\neq_4$. If $\alpha'_{i_0} = 1$, the supports above are bitwise complemented. The same wiring produces the complemented support of $\neq_4$, which is $\neq_4$ itself; no separate complementation gadget is needed.

We have therefore realized $\neq_4$ on the RHS using $\neq_2|\ \mathcal{F}, \delta_0, \delta_1$ gates. Applying Lemma 2.6 completes the reduction. □

We next construct pairwise opposite witnesses so that Lemma 2.7 applies. The following identity controls the sizes of their complementary coordinate blocks.

Lemma 5.5. *Let f be a constraint of arity n and constant Hamming weight k, and let $\alpha, \beta, \gamma \in \mathcal{S}(f)$ be distinct. For $a, b, c \in \{0, 1\}$, let*

$$S^{abc}_{\alpha\beta\gamma} = \{i \in \{1, \ldots, n\} \mid \alpha_i = a, \beta_i = b, \gamma_i = c\}$$

be the set of coordinates in which α, β, γ take the values a, b, c respectively. Then

$$\begin{aligned} |S^{110}_{\alpha\beta\gamma}| - |S^{001}_{\alpha\beta\gamma}| &= |S^{101}_{\alpha\beta\gamma}| - |S^{010}_{\alpha\beta\gamma}| \\ &= |S^{011}_{\alpha\beta\gamma}| - |S^{100}_{\alpha\beta\gamma}| = l_{\alpha\beta\gamma}, \end{aligned}$$

where $l_{\alpha\beta\gamma} = 2k - n - (|S^{111}_{\alpha\beta\gamma}| - |S^{000}_{\alpha\beta\gamma}|)$. Moreover,

$$\begin{aligned} \mathrm{wt}(\mathrm{Min}_3(\alpha, \beta, \gamma)) &= k - 2l_{\alpha\beta\gamma}, \\ \mathrm{wt}(\mathrm{Maj}_3(\alpha, \beta, \gamma)) &= k + l_{\alpha\beta\gamma}. \end{aligned}$$

Proof. Write $d_1 = |S^{110}_{\alpha\beta\gamma}| - |S^{001}_{\alpha\beta\gamma}|$, $d_2 = |S^{101}_{\alpha\beta\gamma}| - |S^{010}_{\alpha\beta\gamma}|$, $d_3 = |S^{011}_{\alpha\beta\gamma}| - |S^{100}_{\alpha\beta\gamma}|$, and $t = |S^{111}_{\alpha\beta\gamma}| - |S^{000}_{\alpha\beta\gamma}|$. Since $\mathrm{wt}(\alpha) = \mathrm{wt}(\beta) = \mathrm{wt}(\gamma) = k$, comparing the numbers of zeros and ones in these strings gives

$$d_1 + d_2 - d_3 + t = d_1 + d_3 - d_2 + t = d_2 + d_3 - d_1 + t = 2k - n.$$

Hence $d_1 = d_2 = d_3 = 2k - n - t = l_{\alpha\beta\gamma}$. Let $\lambda = \mathrm{Min}_3(\alpha, \beta, \gamma)$ and $\mu = \mathrm{Maj}_3(\alpha, \beta, \gamma)$. By the definition of minority,

$$2\,\mathrm{wt}(\lambda) - n = t - d_1 - d_2 - d_3 = 2k - n - 4l_{\alpha\beta\gamma}.$$

Thus $\mathrm{wt}(\lambda) = k - 2l_{\alpha\beta\gamma}$. Summing $\lambda_i + 2\mu_i = \alpha_i + \beta_i + \gamma_i$ over all coordinates gives $\mathrm{wt}(\lambda) + 2\,\mathrm{wt}(\mu) = 3k$, and hence $\mathrm{wt}(\mu) = k + l_{\alpha\beta\gamma}$. □

Define the ternary constraints $h_{1\text{in}3}$ and $h_{2\text{in}3}$ by

$$\mathcal{S}(h_{1\text{in}3}) = \{001, 010, 100\}, \qquad \mathcal{S}(h_{2\text{in}3}) = \{110, 101, 011\}.$$

Lemma 5.6. $\text{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, h_{1\text{in}3}, h_{2\text{in}3})$ *is NP-complete.*

Proof. Take disjoint RHS copies of $h_{1\text{in}3}$ and $h_{2\text{in}3}$; together they realize $h_{1\text{in}3} \otimes h_{2\text{in}3}$. Label the coordinates of the two factors by (x_1, x_2, x_3) and (y_1, y_2, y_3), respectively. Applying the disequality restriction gadget of Lemma 2.5 to each pair (x_i, y_i), $i \in [3]$, realizes a constraint h on the RHS whose support, in the coordinate order $(x_1, x_2, x_3, y_1, y_2, y_3)$, is

$$\mathcal{S}(h) = \{100011, 010101, 001110\}.$$

Thus h is pairwise opposite. The coordinatewise minority and majority of these three support strings are 111000 and 000111, respectively, neither of which belongs to $\mathcal{S}(h)$. Hence h is neither affine nor bijunctive.

The constraint $\neq_2$ is neither 0-valid nor 1-valid, and is neither Horn nor dual-Horn. Therefore, Schaefer's dichotomy (Theorem 2.8) implies that $\text{CSP}(\neq_2, h)$ is NP-complete. Since both $\neq_2$ and h are pairwise opposite, Lemma 2.7 and the construction above give

$$\begin{aligned}\text{CSP}(\neq_2, h) &\leq_T \text{ECSP}(\mathcal{DEQ} \mid \neq_2, h) \\ &\leq_T \text{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, h_{1\text{in}3}, h_{2\text{in}3}).\end{aligned}$$

□

Lemma 5.7. *Let $\mathcal{F}$ be a finite language of constant-weight constraints. If* Min_3 *is neither an upper nor a lower promise polymorphism of $\mathcal{F}$, then one of the following holds:*

1. *There exists a pairwise opposite constraint $h_{\neg\text{aff}}$ that is not affine such that*

$$\text{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1, h_{\neg\text{aff}}) \leq_T \text{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1);$$

2. $\text{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1)$ *is NP-complete.*

Proof. For each fixed witness triple α, β, γ, abbreviate $l = l_{\alpha\beta\gamma}$ and $S^{abc} = S^{abc}_{\alpha\beta\gamma}$. There are two possible obstructions. A minority output may have the correct weight but be missing from the support. If no such output exists, the two failures supply outputs of weight both above and below the required weight, possibly from different constraints. The three strings in each witness triple are pairwise distinct: if two inputs to minority coincide, its output is the third input and cannot witness a failure.

Case 1: a missing output of the correct weight. Suppose $f \in \mathcal{F}$ has weight k and strings $\alpha, \beta, \gamma \in \mathcal{S}(f)$ such that

$$\lambda = \text{Min}_3(\alpha, \beta, \gamma), \qquad \text{wt}(\lambda) = k, \qquad \lambda \notin \mathcal{S}(f).$$

By Lemma 5.5, $l_{\alpha\beta\gamma} = 0$. Pin the coordinates on which all three strings agree. The restricted strings remain feasible, their minority remains infeasible, and their weight differences do not change. For simplicity retain the notation $f, \alpha, \beta, \gamma, \lambda$ for these restrictions.

The remaining coordinates have one of the six nonconstant patterns $001, 010, 100, 110, 101, 011$. Since $l = 0$, each pattern occurs as often as its complement. Match the coordinates in $S^{110}_{\alpha\beta\gamma}$ bijectively with those in $S^{001}_{\alpha\beta\gamma}$, and do the same for the pairs of patterns $(101, 010)$ and $(011, 100)$. Apply the disequality restriction gadget of Lemma 2.5 to every matched coordinate pair; an empty pattern class requires no gadget.

The resulting RHS constraint $h_{\neg\text{aff}}$ is pairwise opposite. Each of α, β, γ satisfies all the added restrictions, so all three remain in its support. Its support is contained in that of f, so λ is still absent. Thus $h_{\neg\text{aff}}$ is not affine. This is the first conclusion of the lemma.

Case 2: outputs on both sides of the required weight. Suppose Case 1 does not occur. Since Min_3 is not a lower promise polymorphism of $\mathcal{F}$, some constraint has a triple whose minority weight exceeds the constraint's weight. Since Min_3 is not an upper promise polymorphism, another triple has minority weight below its constraint's weight. The constraints may coincide. By Lemma 5.5, their respective parameters l are negative and positive.

We describe one construction for either nonzero value of l. Let f and α, β, γ be the corresponding constraint and triple. Pin the coordinates on which all three strings agree, again retaining their names. This leaves the six nonconstant coordinate patterns, and the parameter l is unchanged. The three complementary pattern pairs satisfy

$$|S^{110}| - |S^{001}| = |S^{101}| - |S^{010}| = |S^{011}| - |S^{100}| = l.$$

We compress each complementary pattern pair to a block of $|l|$ equal coordinates so that each of the three witness assignments extends through the gadget. The common imbalance ensures that all three resulting blocks have the same size.

Consider first (S^{110}, S^{001}), of sizes p and q. Put $a = \max\{p, q\}$. Take an RHS copy of $\neq_{2a}$, join p ports of its first equality block to S^{110}, and join q ports of its second equality block to S^{001}, each connection through a separate LHS copy of $\neq_2$. Exactly $|p - q| = |l|$ ports remain dangling, all in one equality block. Repeat this construction for (S^{101}, S^{010}) and (S^{011}, S^{100}).

The resulting constraint f^* has three blocks of $|l|$ coordinates, with equality within each block. If $l < 0$, the three witness strings induce block values $001, 010, 100$. If $l > 0$, they induce $110, 101, 011$. To check the sign, in the first pattern pair the unused ports lie in the first block when $p < q$, giving the complement of pattern 110, namely 001; when $p > q$, they lie in the second block and give the complement of 001, namely 110.

Moreover, f^* has constant Hamming weight. All its RHS constraints do, and each internal LHS disequality consumes exactly one value 1, leaving a fixed number of ones on the dangling ports. Each accepted string is constant on each of the three blocks, and $|l| > 0$. The witness strings therefore fix its weight to $|l|$ if $l < 0$, or $2|l|$ if $l > 0$. In the first case exactly one block must be one; in the second exactly two must be one. The three induced block patterns already exhibit all these possibilities, so they exhaust the support. Thus

$$f^* = \begin{cases} h_{1\text{in}3}^{|l|}, & l < 0, \\ h_{2\text{in}3}^{|l|}, & l > 0, \end{cases}$$

where the superscript denotes coordinate replication. By Lemma 2.4, an instance using the corresponding ternary constraint reduces to one using its replicated version. Substituting the gate for f^* then eliminates that replicated constraint. This makes the ternary constraint available by an instance reduction; it need not be realizable by a three-port gate.

Applying the construction to the negative and positive witnesses makes both ternary constraints available on the RHS. Hence

$$\text{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, h_{1\text{in}3}, h_{2\text{in}3}) \leq_T \text{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1).$$

The source problem is NP-complete by Lemma 5.6, giving the second conclusion. □

Lemma 5.8. *Let $\mathcal{F}$ be a finite language of constant-weight constraints. If* Maj_3 *is neither an upper nor a lower promise polymorphism of $\mathcal{F}$, then one of the following holds:*

1. *There exists a pairwise opposite constraint $h_{\neg\mathrm{bij}}$ that is not bijunctive such that*
$$\mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1, h_{\neg\mathrm{bij}}) \leq_T \mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1);$$
2. $\mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1)$ *is NP-complete.*

Proof. The coordinate constructions in the preceding proof depend on the parameter $l = l_{\alpha\beta\gamma}$, not on whether the witness was obtained from minority or majority. We explain how the majority hypotheses supply the same two cases. Again, the three witness strings must be pairwise distinct, since majority with two equal inputs returns the repeated input.

Suppose first that some constraint f of weight k has $\alpha, \beta, \gamma \in \mathcal{S}(f)$ with
$$\mu = \mathrm{Maj}_3(\alpha, \beta, \gamma), \qquad \mathrm{wt}(\mu) = k, \qquad \mu \notin \mathcal{S}(f).$$
The weight identity gives $l_{\alpha\beta\gamma} = 0$. Pin the coordinates on which the three strings agree and match complementary coordinate patterns exactly as in Case 1 of Lemma 5.7. Disequality restriction produces a pairwise opposite constraint $h_{\neg\mathrm{bij}}$ whose support still contains the restricted witness strings and still excludes their majority. Thus this constraint is not bijunctive and gives the first conclusion.

Otherwise no majority output of the correct weight is missing. Since Maj_3 is not a lower promise polymorphism of $\mathcal{F}$, some output has weight above its constraint's weight; since Maj_3 is not an upper promise polymorphism, another has weight below its constraint's weight. By Lemma 5.5, these witnesses have $l > 0$ and $l < 0$. The compression construction in Case 2 of Lemma 5.7 produces replicated $h_{2\mathrm{in}3}$ and $h_{1\mathrm{in}3}$, respectively. Apply Lemma 2.4 to remove the replications and Lemma 5.6 to obtain NP-completeness. □

Proof of Theorem 5.3. The constraint $\delta_0 \otimes \delta_1$ has support $\{01\}$; it is an even Δ-matroid, and both Min_3 and Maj_3 are upper and lower promise polymorphisms of it. By Lemma 5.1, adding this constraint reduces the tractable cases to Lemma 2.9 and Proposition 4.1.

Suppose none of the tractability conditions holds. By Lemma 5.4,
$$\mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1) \leq_T \mathrm{EO}^c(\mathcal{F}).$$
By Lemma 5.7, either the left-hand problem is already NP-complete, or a pairwise opposite constraint $h_{\neg\mathrm{aff}}$ that is not affine can be added on the RHS without changing its complexity. In the latter case, set $\Gamma = \mathcal{F} \cup \{h_{\neg\mathrm{aff}}\}$. The operation Maj_3 is still neither an upper nor a lower promise polymorphism of Γ, since $\mathcal{F} \subseteq \Gamma$. Applying Lemma 5.8 to Γ either proves NP-hardness, after eliminating $h_{\neg\mathrm{aff}}$, or supplies a pairwise opposite constraint $h_{\neg\mathrm{bij}}$ that is not bijunctive and can also be added on the RHS.

In the remaining case, let $\mathcal{H} = \{\neq_2, h_{\neg\mathrm{aff}}, h_{\neg\mathrm{bij}}\}$. The two witnesses exclude the affine and bijunctive cases, while $\neq_2$ excludes 0-validity, 1-validity, Horn, and dual-Horn. Thus $\mathrm{CSP}(\mathcal{H})$ is NP-complete by Theorem 2.8. Every constraint in $\mathcal{H}$ is pairwise opposite, so Lemma 2.7 and the preceding reductions give
$$\begin{aligned}\mathrm{CSP}(\mathcal{H}) &\leq_T \mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{H}) \\ &\leq_T \mathrm{ECSP}(\mathcal{DEQ} \mid \mathcal{DEQ}, \mathcal{F}, \delta_0, \delta_1) \leq_T \mathrm{EO}^c(\mathcal{F}).\end{aligned}$$
Auxiliary disequalities can be expanded in polynomial time by Lemma 2.6. □

Proof of Theorem 1.1. The tractable cases follow from Lemma 2.9, the constant YES algorithm, and Proposition 4.1. Otherwise, failure of condition 2 gives a NO instance, so Lemma 5.2 yields $\mathrm{EO}^c(\mathcal{F}) \leq_T \mathrm{EO}(\mathcal{F})$. Conditions 1, 3, and 4 also fail, making $\mathrm{EO}^c(\mathcal{F})$ NP-complete by Theorem 5.3. Hence $\mathrm{EO}(\mathcal{F})$ is NP-complete. □

6 Graph Orientations with Unary Pinnings

For an arbitrary Boolean constraint language $\mathcal{F}$, define $\mathrm{GO}(\mathcal{F}) := \mathrm{ECSP}(\neq_2 |\ \mathcal{F})$. Write $\mathcal{U} = \{\delta_0, \delta_1\}$ and

$$\mathrm{GO}^c(\mathcal{F}) := \mathrm{GO}(\mathcal{F}, \mathcal{U}) = \mathrm{ECSP}(\neq_2 |\ \mathcal{F}, \mathcal{U}). \tag{4}$$

Theorem 6.1 (Trichotomy for GO with two pinnings)**.** *Let $\mathcal{F}$ be a fixed finite set of nonzero Boolean constraints.*

1. *If every $f \in \mathcal{F}$ has constant Hamming weight, then $\mathrm{GO}^c(\mathcal{F})$ is polynomial-time solvable if every f is a Δ-matroid, or if some $p \in \{\mathrm{Min}_3, \mathrm{Maj}_3\}$ is an upper or lower promise polymorphism of $\mathcal{F}$. Otherwise it is NP-complete. The constant weight may depend on f.*

2. *If some $f \in \mathcal{F}$ has nonconstant Hamming weight, then*

$$\mathrm{GO}^c(\mathcal{F}) \equiv_T \mathrm{ECSP}(\neq_2, \mathcal{U}, =_1 |\ \mathcal{F}, \mathcal{U}, =_1). \tag{5}$$

   Its classification is inherited from Theorem 6.4: the cases are polynomial-time solvable, NP-complete, or in the residual Δ-matroid parity family. The additional tractable cases covered by [KKR18, Theorem 11] belong to the polynomial-time branch.

Lemma 6.2 (Nonconstant weight permits the free unary)**.** *If $\mathcal{F}$ contains a constraint of nonconstant Hamming weight, then $=_1$ can be added to $\mathrm{GO}^c(\mathcal{F})$ without changing its complexity, and (5) holds.*

Proof. Choose support strings α, β of different weights in some $f \in \mathcal{F}$ with minimum Hamming distance, with $\mathrm{wt}(\alpha) < \mathrm{wt}(\beta)$. Pin their common coordinates and let a be the restriction of α. The remaining support is $\{a, \bar{a}\}$: any third string would form a closer pair of different weights with one of them. Join each one-position of a to a distinct zero-position through $\neq_2$. This realizes $=_d$, where $d = |a| - 2\,\mathrm{wt}(a) > 0$. By Lemma 2.4,

$$\mathrm{ECSP}(\neq_2 |\ \mathcal{F}, \mathcal{U}, =_1) \leq_T \mathrm{ECSP}(\neq_2 |\ \mathcal{F}, \mathcal{U}, =_d) \leq_T \mathrm{GO}^c(\mathcal{F}).$$

Finally, a left unary u can be replaced by the right unary $u(1-x)$ through $\neq_2$, proving (5). □

Theorem 6.3 (Constant-weight GO reduces to EO)**.** *Let $\mathcal{F}$ be a fixed finite language of nonzero constraints, where each $f \in \mathcal{F}$ has arity n_f and constant weight t_f. Define*

$$a_f = \max\{n_f - 2t_f, 0\}, \quad b_f = \max\{2t_f - n_f, 0\}, \quad \widehat{f} = f \otimes \delta_1^{\otimes a_f} \otimes \delta_0^{\otimes b_f},$$

and let $\widehat{\mathcal{F}} = \{\widehat{f} : f \in \mathcal{F}\}$. Then $\widehat{\mathcal{F}}$ is an EO language and

$$\mathrm{GO}^c(\mathcal{F}) \equiv_T \mathrm{EO}^c(\widehat{\mathcal{F}}).$$

Balancing preserves the Δ-matroid condition and the upper and lower promise polymorphisms $\mathrm{Min}_3, \mathrm{Maj}_3$, in both directions.

Proof. At each f-vertex, append the fixed ports of $\widehat{f}$ and attach opposite unary pins to enforce their values. Conversely, split every $\widehat{f}$ into f and its unary factors. These operations preserve satisfying assignments even when loops are present, and $t_f + a_f = (n_f + a_f + b_f)/2$ proves balance. The added coordinates are fixed, so the exchange axiom is unchanged. For either ternary operation p and the fixed suffix c, $p(\alpha c, \beta c, \gamma c) = p(\alpha, \beta, \gamma)c$. Appending c adds a_f to both the output weight and the required weight, preserving whether p is an upper or lower promise polymorphism. □

The *nonzero simulation closures* of a pair of languages consist of all nonzero constraints realizable by bipartite gates on the respective sides.

Theorem 6.4 (Feder–Ford [FF06, Theorem 6.2])**.** *Let $(\mathcal{A}, \mathcal{B})$ be closed under nonzero bipartite simulation and contain $\delta_0, \delta_1, =_1$ on each side. The Schaefer, Δ-matroid intersection, and mixed 2-SAT/Δ-matroid cases specified in that theorem admit polynomial-time algorithms given oracles that decide whether a constraint has a support string after prescribed coordinates are fixed. Every other case is NP-complete unless $\mathcal{A} = \mathcal{B}$ and this common language consists of Δ-matroids and contains binary equality; these remaining cases are Δ-matroid parity problems.*

Proof of Theorem 6.1. For constant weight, apply Theorem 6.3 and the pinned EO dichotomy (Theorem 5.3). Otherwise Lemma 6.2 gives (5). Apply Theorem 6.4 to the nonzero simulation closures of the two constraint languages in this bipartite ECSP. The required restriction queries concern products of original vertex constraints and their coordinatewise upward and downward closures: strings above or below some support string in the coordinatewise order. These queries split into independent queries to fixed support tables. Hardness transfers by substituting the fixed gates for the finitely many simulated constraints used in a reduction. Within the residual family, if every original constraint is coverable in the sense of [KKR18, Definitions 9–10], then [KKR18, Theorem 11] applies: binary disequality and the three unaries are also coverable, and the required even-cover tables can be fixed in advance because the language is fixed. This gives the stated trichotomy. □

For symmetric constraints, this recovers the MIT Hardness Group's dichotomy with constants on general graphs [MAD+26, Theorem 3.12]. A symmetric constraint is a Δ-matroid precisely when its allowed-weight gaps are at most two; maximal step-two intervals give its even covers [KKR18, Definitions 9–10]. The mixed Feder–Ford case requires order-convexity: if $x \le z \le y$ coordinatewise and $x, y \in \mathcal{S}(f)$, then $z \in \mathcal{S}(f)$. For symmetric constraints this forces an interval of allowed weights, hence a Δ-matroid. The left simulation closure contains disequality and every original constraint with all input bits complemented, so the Schaefer branches reduce to languages consisting entirely of affine constraints or entirely of bijunctive constraints.

A Proofs of the Gadget Lemmas

Proof of Lemma 2.4. Let Ω be an instance of ECSP$(\mathcal{F} \mid \mathcal{G}, f)$. Make l disjoint copies of Ω. For each vertex assigned f, merge its l copies into one vertex assigned f^l, placing the l copies of its ith incident edge at the ports in block i. Leave all other vertices unchanged. The resulting instance Ω' uses only $\mathcal{F}$ on the LHS and $\mathcal{G} \cup \{f^l\}$ on the RHS.

A satisfying assignment of Ω extends to Ω' by using the same assignment in every copy. Conversely, the block equalities at each f^l-vertex ensure that a satisfying assignment of Ω' restricts to a satisfying assignment of Ω in each copy. Since l is fixed, the construction takes polynomial time. □

Proof of Lemma 2.5. Place one copy each of f and $\neq_4$ on the RHS. Connect ports i and j of f to ports 1 and 3 of $\neq_4$, respectively, through two copies of $\neq_2$ on the LHS. Use ports 4 and 2 of $\neq_4$ as the external ports corresponding to coordinates i and j, respectively, and leave all other ports of f dangling. If the values at ports i, j of f are a, b, then ports $1, 3$ of $\neq_4$ have values $1-a, 1-b$. Since $\neq_4$ requires its first two inputs to be equal and opposite to its last two, this assignment extends exactly when $a \neq b$. In that case, its ports $4, 2$ have values a, b, respectively. Thus the gate realizes $f_{i\neq j}$ with the original coordinate order. □

Proof of Lemma 2.6. Join one port from each equality block of the RHS constraint $\neq_{2n}$ through $\neq_2$. This yields $\neq_{2n-2}$ on the RHS; repetition yields $\neq_4$ and $\neq_2$.

To increase the arity, connect one port of $\neq_{2k}$ to one port of $\neq_4$, both on the RHS, through an LHS copy of $\neq_2$. The connection fixes the relative values of their two pairs of blocks. Among the remaining ports, $k+1$ have one value and $k+1$ have its complement. After reordering the ports, the gate therefore realizes $\neq_{2k+2}$. Repeating this construction realizes any $\neq_{2k}$ on the RHS, giving

$$\mathrm{ECSP}(\neq_2|\ \mathcal{DEQ}, \mathcal{F}) \leq_T \mathrm{ECSP}(\neq_2|\ \mathcal{F}, \neq_{2n}).$$

To realize $\neq_{2k}$ on the LHS, attach a separate LHS copy of $\neq_2$ to each port of its RHS realization, leaving the other port of each new vertex dangling. This complements every external input, which leaves the support of $\neq_{2k}$ unchanged. These constructions take time polynomial in k, so they give the required reduction for the unbounded family. □

References

[AGH17] Per Austrin, Venkatesan Guruswami, and Johan Håstad. $(2+\varepsilon)$-SAT is NP-hard. *SIAM Journal on Computing*, 46(5):1554–1573, 2017.

[BBKO21] Libor Barto, Jakub Bulín, Andrei Krokhin, and Jakub Opršal. Algebraic approach to promise constraint satisfaction. *Journal of the ACM (JACM)*, 68(4):28:1–28:66, 2021.

[BG19] Joshua Brakensiek and Venkatesan Guruswami. An algorithmic blend of LPs and ring equations for promise CSPs. In *Proceedings of the Thirtieth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 436–455, 2019.

[BG21] Joshua Brakensiek and Venkatesan Guruswami. Promise constraint satisfaction: Algebraic structure and a symmetric Boolean dichotomy. *SIAM Journal on Computing*, 50(6):1663–1700, 2021.

[BGWŽ20] Joshua Brakensiek, Venkatesan Guruswami, Marcin Wrochna, and Stanislav Živný. The power of the combined basic linear programming and affine relaxation for promise constraint satisfaction problems. *SIAM Journal on Computing*, 49(6):1232–1248, 2020.

[BM08] J. A. Bondy and U. S. R. Murty. *Graph Theory*. Springer Publishing Company, Incorporated, 1st edition, 2008.

[Bul17] Andrei A. Bulatov. A dichotomy theorem for nonuniform CSPs. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 319–330, 2017.

[CFS20] Jin-Yi Cai, Zhiguo Fu, and Shuai Shao. Beyond #CSP: A dichotomy for counting weighted Eulerian orientations with ARS. *Information and Computation*, 275:104589, 2020.

[CFX18] Jin-Yi Cai, Zhiguo Fu, and Mingji Xia. Complexity classification of the six-vertex model. *Information and Computation*, 259:130–141, 2018.

[CLX11] Jin-Yi Cai, Pinyan Lu, and Mingji Xia. Computational complexity of Holant problems. *SIAM Journal on Computing*, 40(4):1101–1132, 2011.

[Cor88] Gérard Cornuéjols. General factors of graphs. *Journal of Combinatorial Theory, Series B*, 45(2):185–198, 1988.

[Fed01] Tomás Feder. Fanout limitations on constraint systems. *Theoretical Computer Science*, 255(1–2):281–293, 2001.

[FF06] Tomás Feder and Daniel Ford. Classification of bipartite Boolean constraint satisfaction through delta-matroid intersection. *SIAM Journal on Discrete Mathematics*, 20(2):372–394, 2006. Preprint: ECCC TR05-016.

[GIM03] James F. Geelen, Satoru Iwata, and Kazuo Murota. The linear delta-matroid parity problem. *Journal of Combinatorial Theory, Series B*, 88(2):377–398, 2003.

[GSS26] Jincheng Guan, Shuai Shao, and Ke Shi. An LP algorithm for counting Eulerian orientations through the lens of quasi-polymorphism. *arXiv:2607.27961*, 2026.

[Kar84] Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Combinatorica*, 4(4):373–395, 1984.

[KKR18] Alexandr Kazda, Vladimir Kolmogorov, and Michal Rolínek. Even delta-matroids and the complexity of planar Boolean CSPs. *ACM Transactions on Algorithms (TALG)*, 15(2):22:1–22:33, 2018. Preprint: arXiv:1602.03124.

[MAD+26] MIT Hardness Group, Zachary Abel, Erik D. Demaine, Jenny Diomidova, Jeffery Li, and Zixiang Zhou. Planar graph orientation frameworks, applied to KPlumber and polyomino tiling, 2026. arXiv:2603.03488.

[MW96] Milena Mihail and Peter Winkler. On the number of Eulerian orientations of a graph. *Algorithmica*, 16(4–5):402–414, 1996.

[MWX25] Boning Meng, Juqiu Wang, and Mingji Xia. The $\mathrm{FP}^{\mathrm{NP}}$ versus #P dichotomy for #EO. In *Proceedings of the 57th Annual ACM Symposium on Theory of Computing*, pages 1795–1806, 2025.

[Pau35] Linus Pauling. The structure and entropy of ice and of other crystals with some randomness of atomic arrangement. *Journal of the American Chemical Society*, 57(12):2680–2684, 1935.

[Rys63] Franz Rys. Über ein zweidimensionales klassisches Konfigurationsmodell. *Helvetica Physica Acta*, 36(5):537–559, 1963.

[Sch78] Thomas J. Schaefer. The complexity of satisfiability problems. In *Proceedings of the Tenth Annual ACM Symposium on Theory of Computing*, pages 216–226, 1978.

[Sla41] John C. Slater. Theory of the transition in KH_2PO_4. *The Journal of Chemical Physics*, 9(1):16–33, 1941.

[Zhu20] Dmitriy Zhuk. A proof of the CSP dichotomy conjecture. *Journal of the ACM*, 67(5):30:1–30:78, 2020.