

Independent Set Discovery on Biclique-Free Graphs Is Fixed-Parameter Tractable

Chenghua Liu¹ and Boning Meng²

¹Institute of Software, Chinese Academy of Sciences, Beijing, China

²University of Regensburg, Regensburg, Germany

liuch.russell@gmail.com, mengboning2013@gmail.com

Abstract

INDEPENDENT SET DISCOVERY asks whether a configuration of k tokens on distinct vertices can be transformed into an independent k -set by a sequence of token slides, each moving one token to an unoccupied neighbor; only the terminal configuration must be independent. INDEPENDENT SET DISCOVERY is a central problem in solution discovery: its target is not prescribed and must be chosen together with the token movements needed to reach it. Fellows et al. [6] proved it FPT in k on every fixed bounded-degeneracy class and every nowhere-dense class, leaving the biclique-free case open. The biclique-free setting lies substantially beyond both regimes: biclique-free classes can have unbounded degeneracy and even be somewhere dense. We resolve the open problem affirmatively. Given an n -vertex, m -edge graph promised to be $K_{d,d}$ -free and an initial k -token configuration, our deterministic algorithm computes the minimum number of slides and, in time $2^{O(dk \log k)}(n + m)^{O(1)}$, returns an optimal independent target and a shortest collision-free slide sequence or certifies that no independent target is reachable. Thus the problem is FPT in k for every fixed d and uniformly FPT in $k + d$. The proof combines an exact minimum-cost assignment characterization of token movement with local branching on bounded, cost-relevant *cheap prefixes* of candidate lists. The method also yields an exact FPT algorithm for weighted independent transversals on $K_{d,d}$ -free graphs with overlapping candidate sets, direct exact FPT algorithms for both problems on bounded-degeneracy graphs, an edge-count-sensitive XP algorithm, sharper bounds for bounded s -codegree and unbalanced biclique exclusion, and an exact extension to weighted movement on a separate directed graph.

Acknowledgments

The authors used OpenAI’s ChatGPT in preparing this manuscript, including for language editing and L^AT_EX preparation. ChatGPT also contributed to the exploratory development of most of the arguments. All claims and proofs were independently checked and finalized by the authors, who take full responsibility for the content.

1 Introduction

Solution discovery asks for a feasible solution that is close to a prescribed, possibly infeasible, state. Fellows et al. [6] formalized this viewpoint through reconfiguration moves. In INDEPENDENT SET DISCOVERY, the state is a set S of k occupied vertices of a graph G . A move slides one token along an edge to an unoccupied vertex, and the objective is to reach an independent k -set using at most a given number of moves. Only the terminal configuration

must be independent. This endpoint-only feasibility requirement distinguishes discovery from independent-set reconfiguration and is essential to the algorithm developed here.

The framework captures a natural repair problem: the target is not prescribed, and the cost is the actual movement needed to reach a feasible state. INDEPENDENT SET DISCOVERY is one of the four fundamental problems used by Fellows et al. to introduce solution discovery [6]. It is a basic test case for the framework: pairwise nonadjacency is the canonical packing constraint, yet the target independent set must be chosen jointly with the movement of the tokens. Consequently, endpoint feasibility is coupled to global shortest-path information from the initial configuration.

Since its introduction, solution discovery has developed along several complementary directions. Grobler et al. extended the framework to spanning trees, shortest paths, matchings, and cuts, and classified these problems under several token-movement models [8]. Subsequent work initiated the systematic study of kernelization, including polynomial kernels for INDEPENDENT SET DISCOVERY parameterized by k on nowhere-dense classes [9]. Saito et al. obtained clique-width-based algorithms for INDEPENDENT SET DISCOVERY, hardness results on chordal and diameter-two graphs, and polynomial-time solvability on split graphs [13]. The framework has also led to logical meta-theorems, cost–value objectives, and models separating the feasibility graph from the movement graph [2, 7, 14]. This breadth, and the repeated appearance of INDEPENDENT SET DISCOVERY across these developments, make its structural parameterized complexity a basic question for the emerging theory.

1.1 The biclique-free frontier

Fellows et al. [6] proved that INDEPENDENT SET DISCOVERY is NP-complete on planar graphs of maximum degree four. Parameterized by the combined quantity $k + b$, it remains W[1]-hard even when $C_4, \dots, C_p$ are excluded as induced subgraphs for any fixed p ; parameterized by the slide budget b alone, it is W[1]-hard on 2-degenerate bipartite graphs. On the positive side, they gave an XP algorithm parameterized by treewidth and proved FPT in k on every fixed bounded-degeneracy class and on every nowhere-dense class. Subsequent work broadened the surrounding landscape through kernelization, width-based upper and lower bounds, and logical meta-theorems [9, 13, 2].

Biclique exclusion is a natural and substantially broader frontier. Every a -degenerate graph is $K_{a+1, a+1}$ -free, but the converse fails sharply. For every prime q , the incidence graph of the projective plane $PG(2, q)$ is C_4 -free, $(q + 1)$ -regular, and hence has degeneracy $q + 1$, despite having $2(q^2 + q + 1)$ vertices [11]. Moreover, the 1-subdivision of K_t is C_4 -free for every t , and K_t is its 1-shallow minor; hence the class of C_4 -free graphs is somewhere dense. Thus neither earlier FPT theorem covers the full biclique-free regime. At the same time, excluding a fixed biclique imposes strong extremal structure [10]. A positive result would therefore show that this extremal sparsity alone suffices for fixed-parameter tractability, well beyond the previously understood sparse classes.

There is also a concrete obstacle to extending the earlier algorithm. It constructs a k -independence-covering family—a family $\mathcal{F}$ of independent sets such that every independent set of size at most k is contained in some member of $\mathcal{F}$ —and solves one minimum-cost matching instance for each member [6, 12]. Kuhn, Lokshantov, and Miller proved that, for every function f and every fixed $\varepsilon > 0$, no bound of the form $f(k)n^{k/4-\varepsilon}$ holds for such families on all C_4 -free graphs [11]. Since $C_4 = K_{2,2}$, explicitly enumerating these global families cannot yield an FPT algorithm even in the first nontrivial biclique-free case. An FPT resolution therefore cannot proceed by explicitly enumerating such a global covering family and needs a different mechanism for representing candidate targets.

Against this background, Fellows et al. explicitly posed the following question [6].

Open Problem 1.1 (Fellows et al. [6]). For every fixed integer $d \geq 2$, is INDEPENDENT SET DISCOVERY fixed-parameter tractable parameterized by the number k of tokens on $K_{d,d}$ -free graphs?

1.2 Our results

For an initial configuration S , let $\text{OPT}(G, S)$ be the minimum number of slides needed to reach an independent set of size $|S|$, and put $\text{OPT}(G, S) = +\infty$ if no such set is reachable. Our main result resolves Theorem 1.1 and strengthens the requested result in two ways: the dependence is uniform in $k + d$, and the algorithm computes the optimum together with an optimal slide sequence.

Theorem 1.2 (Main theorem). *Let $d, k \geq 2$ be integers. Given an n -vertex, m -edge graph G promised to be $K_{d,d}$ -free and a set $S \in \binom{V(G)}{k}$, one can deterministically compute $\text{OPT}(G, S)$. If $\text{OPT}(G, S) < +\infty$, the algorithm returns an independent k -set T and a collision-free sequence of exactly $\text{OPT}(G, S)$ token slides from S to T ; otherwise, it reports that no independent k -set is reachable from S . Its running time is*

$$2^{O(dk \log k)}(n + m)^{O(1)}.$$

Consequently, the budgeted decision version is FPT in k for every fixed d and uniformly FPT in $k + d$. It is obtained by comparing $\text{OPT}(G, S)$ with the binary-encoded budget b , which adds only polynomial dependence on the encoding length of b .

The proof yields three further contributions.

1. *A general weighted static theorem.* We solve a weighted independent-transversal problem in which the k candidate sets may overlap and the cost of a vertex depends on its label. Theorem 4.4 gives the exact optimization algorithm, and Theorem 3.3 transfers both the optimum and a witness back to INDEPENDENT SET DISCOVERY.
2. *A direct proof for bounded-degeneracy graphs.* The same search gives a short direct proof of FPT on bounded-degeneracy classes. In contrast to the covering-family proof, it uses only a local edge count and prefix branching. It also gives an explicit uniform running time in the number of tokens and the degeneracy, works for the weighted problem, and yields exact optimization and sequence reconstruction. Thus the method recovers the earlier tractability theorem while strengthening its algorithmic formulation; see Theorem 5.3.
3. *Further algorithms and sharper bounds.* The local conflict bound yields an edge-count-sensitive XP algorithm on arbitrary graphs and sharper uniform FPT bounds for bounded s -codegree (the maximum common-neighborhood size of an s -set) and unbalanced $K_{s,t}$ exclusion. The latter bound depends exponentially on the smaller side s , whereas applying the balanced theorem with $d = t$ would put the larger side in the extremal exponent. Finally, the assignment argument extends to a two-graph model with a biclique-free feasibility graph and an arbitrary directed, nonnegatively weighted movement graph. These consequences are stated in Theorems 5.2 and 5.4 to 5.6.

1.3 Proof overview

The proof separates movement from terminal feasibility. Temporarily label the tokens by their starting positions. For any fixed target T , the minimum number of slides needed to reach T equals the cost of a minimum assignment of starting positions to vertices of T , where

assignment costs are shortest-path distances. A simple exchange argument handles occupied vertices along the assigned paths: whenever a token blocks another token's path, the two tokens exchange destinations in the analysis. The assignment cost does not increase, and every actual slide still enters an unoccupied vertex. Thus the dynamic problem reduces exactly to selecting pairwise nonadjacent destinations of minimum total assignment cost.

We formulate that static task as a weighted independent-transversal problem. Each starting token is a coordinate, its reachable vertices are possible destinations, and two selected destinations conflict if they are equal or adjacent. The destination lists may overlap and costs depend on the coordinate, so a direct application of a standard transversal theorem is not enough.

The algorithm uses a *cheap-prefix dichotomy*. Suppose that r labels remain. For each label, form a prefix of its $(dr)^{O(d)}$ cheapest destinations compatible with the partial assignment. If some residual candidate list is shorter than this prefix length, we branch on its entire list. Otherwise all prefixes are full. If the sum of their maximum costs fits the residual budget, the Kővári–Sós–Turán bound and the independent-transversal criterion of Wanless and Wood [10, 15] guarantee compatible representatives. If this sum exceeds the budget, every feasible solution must select a vertex from at least one prefix, because a solution avoiding all prefixes would have cost at least the sum of their coordinate-wise lower bounds. Branching over the prefixes is therefore exhaustive. Each branch fixes one label, so the depth is at most k and the branching factor depends only on $k + d$.

The decisive simplification is locality. Extremal sparsity is invoked only on bounded, cost-relevant prefixes, rather than through a global family that must cover every small independent set. This is precisely why the method continues to work in the C_4 -free setting where small global covering families need not exist.

Viewed statically, our problem is a graph-metric instance of *Dispersion* in the movement-minimization framework [4, 5, 1]. Our contribution in this setting is an exact FPT algorithm under biclique-free terminal conflicts, allowing overlapping destination lists and label-dependent costs and reconstructing an optimal token-slide sequence.

Organization. Section 2 defines the discovery model and the weighted static problem. Section 3 proves the assignment-distance equivalence. Section 4 then gives the biclique-free cheap-prefix algorithm and completes the proof of Theorem 1.2. Finally, Section 5 develops the additional sparse-graph and weighted-movement consequences.

2 Preliminaries

We first formalize the discovery model and then define the weighted static problem used by the algorithm.

2.1 Configurations and Independent Set Discovery

Definition 2.1 (Configuration and token slide). Let G be a finite simple undirected graph and let k be a nonnegative integer. A k -token configuration is a set $Q \subseteq V(G)$ with $|Q| = k$. Its vertices are *occupied*. For two k -token configurations Q and Q' , write $Q \rightarrow_G Q'$ if there are $u \in Q$ and $v \in V(G) \setminus Q$ such that $uv \in E(G)$ and

$$Q' = (Q \setminus \{u\}) \cup \{v\}.$$

The transition $Q \rightarrow_G Q'$ is a *token slide*. A *collision-free slide sequence* of length ℓ is a sequence $Q_0, Q_1, \dots, Q_\ell$ of k -token configurations such that $Q_{j-1} \rightarrow_G Q_j$ for every $j \in \{1, \dots, \ell\}$.

Configurations are sets, so a slide must enter an unoccupied vertex. The tokens are unlabeled; labels will be introduced only as an algorithmic device for assigning starting positions to destinations.

Definition 2.2 (Independent Set Discovery). An instance of INDEPENDENT SET DISCOVERY is a triple (G, S, b) , where G is a finite simple undirected graph, $S \subseteq V(G)$ is an initial configuration, and b is a nonnegative integer encoded in binary. Let $k = |S|$. The instance is a yes-instance if there exist an integer ℓ satisfying $0 \leq \ell \leq b$, an independent k -token configuration T , and a collision-free slide sequence

$$S = Q_0, Q_1, \dots, Q_\ell = T.$$

Only the terminal configuration $Q_\ell = T$ is required to be independent. The optimization version omits b and, given (G, S) , asks for the minimum possible ℓ ; its value is $\text{OPT}(G, S)$, with $\text{OPT}(G, S) = +\infty$ if no independent k -set is reachable from S .

2.2 Numbers, graphs, and distances

We write $\mathbb{N} = \{1, 2, \dots\}$, $\mathbb{N}_0 = \mathbb{N} \cup \{0\}$, and $[q] = \{1, \dots, q\}$ for $q \in \mathbb{N}$. All graphs are finite, simple, and undirected unless a directed graph is explicitly mentioned. For a graph G , its vertex and edge sets are $V(G)$ and $E(G)$; we usually write $n = |V(G)|$ and $m = |E(G)|$. A set $I \subseteq V(G)$ is *independent* if no edge of G has both endpoints in I .

For $X \subseteq V(G)$, let

$$N_G(X) = \{v \in V(G) \setminus X : uv \in E(G) \text{ for some } u \in X\}, \quad N_G[X] = X \cup N_G(X).$$

For a single vertex, we write $N_G(v)$ and $N_G[v]$. The subgraph induced by X is denoted by $G[X]$. For disjoint $X, Y \subseteq V(G)$, let $e_G(X, Y)$ be the number of edges with one endpoint in X and the other in Y . Graph subscripts are omitted when the underlying graph is clear.

A path has length equal to its number of edges. For $u, v \in V(G)$, the distance $\text{dist}_G(u, v)$ is the length of a shortest u - v path, and is defined to be $+\infty$ when u and v lie in different connected components. In an unweighted graph, all distances from one source, together with corresponding shortest-path predecessors, can be computed by breadth-first search in $O(n + m)$ time.

For $d \in \mathbb{N}$, $K_{d,d}$ is the complete bipartite graph with d vertices in each part. A graph is $K_{d,d}$ -free when it has no subgraph isomorphic to $K_{d,d}$. This is stronger than forbidding an induced $K_{d,d}$. Edges within either side of the biclique are immaterial because the forbidden copy need not be induced.

For our biclique-free algorithms, d is part of the input and $K_{d,d}$ -freeness is a promise: the algorithms need not verify it.

2.3 Independent transversals and overlapping candidate sets

Let $C_1, \dots, C_r$ be pairwise disjoint vertex sets in a graph H . An *independent transversal* is an independent set containing exactly one vertex from each C_i . We will use a variant in which each label i has a *candidate set* A_i of admissible vertices, and these candidate sets are not required to be disjoint. The intended interpretation is that label i represents a token and A_i contains its possible destinations; the same vertex may therefore belong to the candidate sets of several tokens. A valid selection chooses one vertex $x_i \in A_i$ for each label so that the selected vertices are distinct and pairwise nonadjacent. The following definition adds coordinate-dependent costs and a budget to this version with overlapping candidate sets.

Definition 2.3 (Weighted independent transversal). For $k \in \mathbb{N}$, an instance $\mathcal{I} = (G, (A_i, c_i)_{i \in [k]}, B)$ of the decision version of WEIGHTED INDEPENDENT TRANSVERSAL consists of

- a graph G ;
- an explicitly listed finite candidate set $A_i \subseteq V(G)$ for every label $i \in [k]$, where candidate sets may overlap;
- a nonnegative rational cost $c_i(v)$ for every $v \in A_i$; and
- a nonnegative rational budget B .

A *feasible transversal* is a tuple $(x_1, \dots, x_k)$ with $x_i \in A_i$ for every i , all entries distinct, and no two entries adjacent in G . Its cost is $\sum_{i=1}^k c_i(x_i)$. The decision problem asks whether a feasible transversal of cost at most B exists. An optimization instance omits B and asks for a minimum-cost feasible transversal, or for a report that none exists. We write $|\mathcal{I}|$ for the total binary encoding length in either version. We use *label* and *coordinate* interchangeably for an index $i \in [k]$.

All integers and rational costs in a WEIGHTED INDEPENDENT TRANSVERSAL instance are encoded in binary; the same holds for the budget B when it is present. Arithmetic operations and comparisons have polynomial bit complexity.

In the reduction from INDEPENDENT SET DISCOVERY, A_i will contain the vertices reachable from the initial position s_i , with $c_i(v) = \text{dist}_G(s_i, v)$.

3 From token slides to weighted transversals

To reduce INDEPENDENT SET DISCOVERY to WEIGHTED INDEPENDENT TRANSVERSAL, we temporarily label the tokens and assign one destination to each label. We now prove that the resulting static assignment cost is exactly the number of slides needed by the original unlabeled tokens. The key freedom is that, if one token blocks another token's assigned shortest path, the two tokens may exchange their assigned destinations.

All costs in this section are interpreted in the extended nonnegative integers. We use the conventions

$$a + (+\infty) = +\infty \quad \text{and} \quad \min \emptyset = +\infty.$$

For two sets $Q, T \subseteq V(G)$ of equal cardinality, define their *assignment distance*

$$\mu_G(Q, T) = \min_{\pi: Q \rightarrow T \text{ bijective}} \sum_{u \in Q} \text{dist}_G(u, \pi(u)).$$

Equivalently, $\mu_G(Q, T)$ is the cost of a minimum-weight perfect matching between the starting vertices Q and target vertices T , where the edge ut has weight $\text{dist}_G(u, t)$. An infinite weight means that the two vertices lie in different connected components. The following matching characterization is known in closely related unlabeled-pebble models [3, 6]. The proof is constructive. Its central idea is to follow an assigned shortest path only to its first unoccupied vertex. If another token blocks that path, the two tokens exchange their assigned destinations in the analysis; this exchange is bookkeeping and does not contribute to the number of slides.

Lemma 3.1 (Assignment-distance lemma). *Let G be a finite simple undirected graph, and let $Q, T \subseteq V(G)$ have equal cardinality. The minimum length of a collision-free token-slide sequence from Q to T equals $\mu_G(Q, T)$, where this minimum is $+\infty$ if no such sequence exists; intermediate configurations are unrestricted beyond collision-freedom.*

Proof. Temporarily label every token by its initial vertex. Any slide sequence induces a bijection $\pi : Q \rightarrow T$. A token starting at u traverses at least $\text{dist}_G(u, \pi(u))$ edges, so every sequence has length at least $\mu_G(Q, T)$. In particular, no sequence exists when $\mu_G(Q, T) = +\infty$.

Suppose that $\mu_G(Q, T) < +\infty$. We prove by induction on $\mu_G(Q, T)$ that a sequence of exactly $\mu_G(Q, T)$ slides exists. The case $\mu_G(Q, T) = 0$ is immediate. Otherwise $Q \neq T$. Choose an unoccupied target $t \in T \setminus Q$, fix an optimal bijection $\pi : Q \rightarrow T$, and let $s = \pi^{-1}(t)$ be the token assigned to t . Along a shortest s - t path, let v be the first unoccupied vertex and let u be its predecessor. Such a vertex exists because $t \notin Q$. By the choice of v , the predecessor u is occupied and $uv \in E(G)$.

If $u \neq s$, let $t_u = \pi(u)$ and exchange the destinations assigned to the tokens at s and u . Because u lies on a shortest s - t path, the triangle inequality gives

$$\begin{aligned} \text{dist}_G(u, t) + \text{dist}_G(s, t_u) &\leq \text{dist}_G(u, t) + \text{dist}_G(s, u) + \text{dist}_G(u, t_u) \\ &= \text{dist}_G(s, t) + \text{dist}_G(u, t_u). \end{aligned}$$

Thus the exchanged assignment costs no more than the optimal assignment π , and hence it is also optimal. If $u = s$, no exchange is needed. In either case, we now have an optimal assignment that sends the token at u to t .

Slide that token from u to the empty neighbor v , and write $Q' = (Q \setminus \{u\}) \cup \{v\}$. The suffix of the chosen path from v to t is shortest, so replacing source u by source v in the current assignment gives

$$\mu_G(Q', T) \leq \mu_G(Q, T) - 1.$$

Conversely, take an optimal assignment from Q' to T and replace its source v by the adjacent source u , keeping the same destination. The triangle inequality gives

$$\mu_G(Q, T) \leq 1 + \mu_G(Q', T).$$

Therefore $\mu_G(Q', T) = \mu_G(Q, T) - 1$. The induction hypothesis transforms Q' into T in that many further slides. Prepending the slide $u \rightarrow v$ produces an optimal sequence from Q to T . Every actual slide enters an unoccupied vertex, so the constructed sequence is collision-free. $\square$

Corollary 3.2 (Static endpoint formulation). *Let (G, S, b) be an instance of INDEPENDENT SET DISCOVERY, where $S = \{s_1, \dots, s_k\}$, and define $c_i : V(G) \rightarrow \mathbb{N}_0 \cup \{+\infty\}$ by $c_i(v) = \text{dist}_G(s_i, v)$. The instance is feasible if and only if there are distinct, pairwise nonadjacent vertices $x_1, \dots, x_k \in V(G)$ such that*

$$\sum_{i=1}^k c_i(x_i) \leq b.$$

Moreover,

$$\text{OPT}(G, S) = \min_{\substack{x_1, \dots, x_k \in V(G) \text{ distinct} \\ \{x_i, x_j\} \notin E(G) \ (i \neq j)}} \sum_{i=1}^k c_i(x_i). \quad (1)$$

Proof. Labeling tokens by their starting vertices, every slide sequence ending at an independent target yields a feasible tuple whose assignment cost is at most the sequence length. Conversely, a feasible tuple defines the independent set $T = \{x_1, \dots, x_k\}$, and the assignment $s_i \mapsto x_i$ gives $\mu_G(S, T) \leq \sum_i c_i(x_i)$. Theorem 3.1 then yields a slide sequence of length at most this sum. Taking minima proves both assertions. $\square$

Corollary 3.3 (Algorithmic transfer from WEIGHTED INDEPENDENT TRANSVERSAL to INDEPENDENT SET DISCOVERY). *Let $\mathcal{C}_p$ be a promise class of graphs indexed by a parameter p . Suppose that, given any WEIGHTED INDEPENDENT TRANSVERSAL optimization instance $\mathcal{I}$ with k candidate sets and underlying graph in $\mathcal{C}_p$, a minimum-cost transversal or a report of nonexistence can be computed in time*

$$f(k, p)|\mathcal{I}|^{O(1)}.$$

Then, given a graph $G \in \mathcal{C}_p$ and a set $S \in \binom{V(G)}{k}$, one can compute $\text{OPT}(G, S)$ in time

$$f(k, p)(n + m)^{O(1)}.$$

If the optimum is finite, the algorithm also returns an optimal independent target and a shortest collision-free slide sequence; otherwise, it reports nonexistence. Here $n = |V(G)|$ and $m = |E(G)|$. The statement remains valid when p is an input-dependent quantity such as $m = |E(G)|$.

Proof. Write $S = \{s_1, \dots, s_k\}$. For label i , take as its candidate set the vertices in the connected component of s_i , and give candidate v cost $c_i(v) = \text{dist}_G(s_i, v)$. These distances and the candidate sets are obtained by k breadth-first searches. By Theorem 3.2, the optimum of this WEIGHTED INDEPENDENT TRANSVERSAL instance is exactly $\text{OPT}(G, S)$.

If the WEIGHTED INDEPENDENT TRANSVERSAL algorithm reports that no transversal exists, then Theorem 3.2 gives $\text{OPT}(G, S) = +\infty$, and we stop. Otherwise, let $(x_1, \dots, x_k)$ be an optimal tuple and $T = \{x_1, \dots, x_k\}$. Its pairing $s_i \mapsto x_i$ is a minimum-cost matching between S and T : a cheaper bijection to the same target would be a cheaper feasible tuple. The constructive argument in Theorem 3.1 can therefore be implemented to output exactly $\text{OPT}(G, S)$ slides. For an explicit polynomial implementation, precompute all-pairs unweighted distances and one next edge on every finite shortest path. In each iteration, scan the current assigned shortest path until its first unoccupied vertex, perform the destination exchange from the proof if needed, and make one slide. Every assigned path is simple, so the scan uses at most $n - 1$ edges; moreover any finite optimum is at most $k(n - 1)$. Thus distance computation and sequence reconstruction add only a polynomial factor. $\square$

4 The cheap-prefix algorithm

We first derive a biclique-free transversal certificate and then use it in the cheap-prefix recursion for WEIGHTED INDEPENDENT TRANSVERSAL.

4.1 Sparse conflicts and the threshold

Before defining the recursion, we prove the structural statement that makes its bounded branching possible. Suppose that each of r labels has a candidate set of t vertices. We want to select one vertex from every candidate set so that the selected vertices are distinct and pairwise nonadjacent. The candidate sets may overlap, so both equality and adjacency count as conflicts.

We next translate the Kővári–Sós–Turán bound [10] into a bound on conflicts between two possibly overlapping candidate sets. Fix an integer $d \geq 2$, a $K_{d,d}$ -free graph G , and an integer $t \geq 1$. Let $X, Y \subseteq V(G)$ satisfy $|X| = |Y| = t$, and form disjoint labeled copies

$$X^L = \{x^L : x \in X\}, \quad Y^R = \{y^R : y \in Y\}.$$

Choosing one vertex from each candidate set produces one *cross-pair* in $X^L \times Y^R$. Hence there are exactly

$$|X^L \times Y^R| = t^2$$

possible cross-pairs. Pairs with both endpoints in the same copied set are irrelevant, since the two choices come from different candidate sets.

Let $J_{X,Y}$ be the bipartite graph with parts X^L and Y^R in which

$$x^L y^R \in E(J_{X,Y}) \iff xy \in E(G).$$

Thus the edges of $J_{X,Y}$ are precisely the adjacency-conflict cross-pairs. The graph $J_{X,Y}$ is $K_{d,d}$ -free. Indeed, a $K_{d,d}$ in $J_{X,Y}$ would project to two sets of d distinct vertices of G . These two sets must be disjoint, since a vertex occurring on both sides would require a loop in G . Their projections would therefore form a $K_{d,d}$ in G .

The balanced Kővári–Sós–Turán bound consequently gives

$$|E(J_{X,Y})| \leq (d-1)^{1/d} t^{2-1/d} + (d-1)t. \quad (2)$$

There is one additional equality-conflict cross-pair $x^L x^R$ for each $x \in X \cap Y$. Its endpoints are distinct copied vertices lying in different bipartition classes, so this does not violate bipartiteness. Since $|X \cap Y| \leq t$, the total number of adjacency-or-equality conflict cross-pairs is at most

$$(d-1)^{1/d} t^{2-1/d} + (d-1)t + t.$$

Accordingly, define

$$p_d(t) = (d-1)^{1/d} t^{-1/d} + \frac{d}{t}.$$

The proportion of the t^2 cross-pairs that are conflicts is then at most

$$\frac{(d-1)^{1/d} t^{2-1/d} + (d-1)t + t}{t^2} = p_d(t).$$

We also use the following block-average crossing-degree theorem. If $C_1, \dots, C_r$ is a partition of a graph, an edge is *crossing at C_i* when it has exactly one endpoint in C_i .

Theorem 4.1 (Wanless–Wood criterion [15]). *Let H be a finite graph, let $r, t \in \mathbb{N}$, and let $(C_1, \dots, C_r)$ be a partition of $V(H)$ with $|C_i| \geq t$ for every $i \in [r]$. If*

$$e_H(C_i, V(H) \setminus C_i) \leq \frac{t}{4} |C_i| \quad \text{for every } i \in [r],$$

then H has an independent transversal of $(C_1, \dots, C_r)$.

Lemma 4.2 (Biclique-free transversal lemma). *Let $d, r, t \in \mathbb{N}$ with $d, r \geq 2$, let G be $K_{d,d}$ -free, and let $B_1, \dots, B_r \subseteq V(G)$ be possibly overlapping sets of the same size t . If*

$$4(r-1)p_d(t) < 1, \quad (3)$$

then there are representatives $y_i \in B_i$ that are distinct and pairwise nonadjacent.

Proof. For every i , make a disjoint copy $C_i = \{(i, v) : v \in B_i\}$. Define an r -partite conflict graph H on $C_1 \dot{\cup} \dots \dot{\cup} C_r$ by joining (i, u) and (j, v) , for $i \neq j$, exactly when $u = v$ or $uv \in E(G)$. Thus an independent transversal of the copied blocks is exactly a tuple of distinct, pairwise nonadjacent representatives of the original candidate sets. Notice that the copy (i, v) records both the label i and the original vertex v ; the copies are new vertices of H , not additional vertices of G .

Fix $i < j$. By the projection argument accompanying (2), the subgraph between C_i and C_j containing only conflicts of the form $uv \in E(G)$ is $K_{d,d}$ -free. By (2), there are at most $(d-1)^{1/d}t^{2-1/d} + (d-1)t$ such edges. Equality contributes at most $|B_i \cap B_j| \leq t$ further edges, so

$$e_H(C_i, C_j) \leq p_d(t)t^2. \quad (4)$$

For each i , the number of edges crossing at C_i is

$$\sum_{j \neq i} e_H(C_i, C_j) \leq (r-1)p_d(t)t^2 < \frac{t^2}{4} = \frac{t}{4}|C_i|.$$

The last strict inequality is exactly (3). The Wanless–Wood criterion therefore gives an independent transversal of $C_1, \dots, C_r$, which projects to the required representatives. $\square$

Lemma 4.3 (Explicit prefix threshold). *For integers $d, r \geq 2$, define*

$$M(d, r) = (d-1)(4(r-1))^d + 4d^2(r-1). \quad (5)$$

Then

$$4(r-1)p_d(M(d, r)) < 1.$$

Proof. Put $Q = 4(r-1)$ and set

$$t = M(d, r) = (d-1)Q^d + d^2Q, \quad A = (d-1)^{1/d}Q, \quad x = t^{1/d}.$$

Since $t = x^d$ and $A = (d-1)^{1/d}Q$, we have

$$4(r-1)p_d(t) = Qp_d(t) = \frac{A}{x} + \frac{dQ}{x^d}.$$

Thus it remains to prove that the last expression is less than one. We have $x > A$ and $x^d - A^d = d^2Q$. Factoring the difference of powers yields

$$d^2Q = (x - A) \sum_{j=0}^{d-1} x^{d-1-j} A^j < dx^{d-1}(x - A).$$

Consequently $x^{d-1}(x - A) > dQ$, and division by $x^d = t$ gives

$$\frac{A}{x} + \frac{dQ}{x^d} < 1.$$

The left-hand side is $Qp_d(t)$, proving the certificate. $\square$

For integers $d, r \geq 2$ and a $K_{d,d}$ -free graph, the two lemmas imply that arbitrary $M(d, r)$ -element subsets of the r candidate sets admit compatible representatives. For fixed d , $M(d, r) = \Theta_d(r^d)$.

4.2 The cheap-prefix certificates

A naive algorithm could choose a destination for each of the k labels and test the resulting k -tuple. This takes roughly n^k time. The goal is to replace the factor n by a bound depending only on d and the number r of labels that remain.

Think of a recursive state as a partial solution. Its *residual budget* is the original budget minus the costs already paid, and the *residual candidate set* of an unassigned label consists of the vertices that do not conflict with an already chosen destination and whose individual

cost does not exceed the residual budget. The formal state below records exactly these two objects.

Fix an arbitrary total order $\prec$ on $V(G)$, used only to break ties between vertices of equal cost. Order each residual candidate set by nondecreasing coordinate cost, breaking ties according to $\prec$. The first

$$\min\{M(d, r), |A_i|\}$$

vertices of the residual candidate set A_i form the label's *cheap prefix*. A cheap prefix is *full* if it contains exactly $M(d, r)$ vertices. The tie-breaking rule makes each prefix uniquely defined but has no effect on the correctness of the algorithm. Exactly one of the following situations occurs.

- (i) *A residual candidate set is small.* It has fewer than $M(d, r)$ vertices, so we branch over all vertices in that candidate set and thereby assign the label.
- (ii) *All prefixes are full and collectively affordable.* If the sum of their largest costs is at most the residual budget, then every selection from the prefixes is affordable. By [Theorems 4.2 and 4.3](#), at least one such selection is distinct and pairwise nonadjacent. The decision procedure may therefore return yes without searching below this state.
- (iii) *All prefixes are full but collectively too expensive.* A feasible solution cannot avoid every prefix: outside a prefix, the cost for that coordinate is at least the prefix maximum, and these lower bounds already exceed the budget. Hence, in every feasible solution, at least one label is assigned a vertex from its cheap prefix. We branch on every possible such label-vertex pair.

Thus Case (ii) certifies existence, whereas Case (iii) certifies an exhaustive branching rule; neither case makes a coordinate-wise greedy choice.

Theorem 4.4 (Weighted transversal theorem). *Let $d, k \geq 2$ be integers, and let $\mathcal{I}$ be a WEIGHTED INDEPENDENT TRANSVERSAL instance with k candidate sets whose underlying graph is promised to be $K_{d,d}$ -free. The decision version can be solved deterministically, with a feasible tuple returned for every yes-instance, in time*

$$O\left((kM(d, k))^k |\mathcal{I}|^{O(1)}\right) = 2^{O(dk \log k)} |\mathcal{I}|^{O(1)}. \quad (6)$$

The optimization version can be solved within the same parameterized bound: the algorithm returns a globally minimum-cost transversal or reports that no transversal exists.

The remainder of this section defines the recursion and proves this theorem.

4.3 The recursive state

For the decision version, fix a WEIGHTED INDEPENDENT TRANSVERSAL instance $\mathcal{I}$ with original candidate sets $A_1^0, \dots, A_k^0$, costs c_i , and original budget B_0 . A recursive state stores

- a set $D_F \subseteq [k]$ of already assigned labels;
- a partial assignment $F : D_F \rightarrow V(G)$;
- the unassigned labels $L = [k] \setminus D_F$;
- the residual budget B ; and
- a residual candidate set A_i for every $i \in L$.

Write $X_F = F(D_F) = \{F(i) : i \in D_F\}$ for the set of vertices already chosen. The state maintains

$$F(i) \in A_i^0 \ (i \in D_F), \quad F \text{ is injective, and } X_F \text{ is independent,} \quad (\text{I1})$$

$$B = B_0 - \sum_{i \in D_F} c_i(F(i)), \quad (\text{I2})$$

$$A_j = \{u \in A_j^0 \setminus N[X_F] : c_j(u) \leq B\} \quad (j \in L). \quad (\text{I3})$$

Invariant (I3) has two purposes. Deleting $N[X_F]$ prevents a future equality or adjacency conflict with a chosen vertex. Deleting a vertex of individual cost greater than B is safe because all costs are nonnegative. Membership in A_j certifies only that a single choice is affordable, not that arbitrary choices from all residual candidate sets fit the budget together. At the root, $F = \emptyset$, $L = [k]$, $B = B_0$, and $A_i = \{v \in A_i^0 : c_i(v) \leq B_0\}$.

4.4 The cheap-prefix dichotomy

The next lemma states the branching argument formally. Its affordable-prefix part uses the transversal certificate already established in Theorems 4.2 and 4.3; its over-budget part is a sorted-cost lower bound.

At the current state, a *feasible completion* is a tuple $(x_i)_{i \in L}$ with $x_i \in A_i$ for every $i \in L$, with distinct and pairwise nonadjacent entries, and with total residual cost at most B .

Lemma 4.5 (Cheap-prefix dichotomy). *Let $d \geq 2$ be an integer. Consider a state with $r = |L| \geq 2$ in a $K_{d,d}$ -free graph, and suppose $|A_i| \geq M(d, r)$ for every $i \in L$. Let P_i be the first $M(d, r)$ vertices of A_i in the fixed order extending nondecreasing cost, and put*

$$a_i = \max_{v \in P_i} c_i(v).$$

Then the following statements hold.

- (a) *If $\sum_{i \in L} a_i \leq B$, there are distinct, pairwise nonadjacent vertices $y_i \in P_i$ for $i \in L$ such that $\sum_{i \in L} c_i(y_i) \leq B$.*
- (b) *If $\sum_{i \in L} a_i > B$, every feasible completion $(x_i)_{i \in L}$ has $x_i \in P_i$ for at least one $i \in L$.*

Proof. For (a), Theorem 4.3 gives $4(r-1)p_d(M(d, r)) < 1$. Applying Theorem 4.2 to the sets $(P_i)_{i \in L}$ supplies distinct, pairwise nonadjacent representatives $y_i \in P_i$. Each satisfies $c_i(y_i) \leq a_i$, so their total cost is at most B .

For (b), suppose a feasible completion avoids every prefix. The ordering is nondecreasing in cost, so $c_i(x_i) \geq a_i$ for every $i \in L$, including when costs tie. Therefore $\sum_{i \in L} c_i(x_i) \geq \sum_{i \in L} a_i > B$, a contradiction. $\square$

When a branch fixes $F(i) = v$, it spends $c_i(v)$, removes label i , and updates every remaining candidate set to

$$A'_j = \{u \in A_j \setminus N[v] : c_j(u) \leq B'\}, \quad B' = B - c_i(v). \quad (7)$$

Closed-neighborhood deletion simultaneously enforces distinctness and nonadjacency. Since the branch chooses $v \in A_i \subseteq A_i^0$, the new assignment is also legal for label i . Thus the update preserves (I1)–(I3).

4.5 The decision procedure

The following box is the complete recursive solver. “Branch on (i, v) ” means create the child in (7); return yes if some child returns yes, and return no if every child returns no.

Cheap-prefix decision procedure at a state $(F, L, B, (A_i)_{i \in L})$. 1. Return no if $B < 0$ or some A_i is empty. Return yes if $L = \emptyset$ or $L = \{i\}$. 2. Put $r = L$. For each $i \in L$, let P_i be the first $\min\{M(d, r), A_i \}$ vertices of A_i in nondecreasing cost order. 3. If some $A_i < M(d, r)$, choose one such i and branch on every (i, v) with $v \in A_i = P_i$. 4. Otherwise set $a_i = \max_{v \in P_i} c_i(v)$ for every $i \in L$. If $\sum_{i \in L} a_i \leq B$, return yes. 5. If $\sum_{i \in L} a_i > B$, branch on every (i, v) with $i \in L$ and $v \in P_i$.

Step 1 is valid for a single remaining label because its nonempty residual candidate set already contains only vertices compatible with X_F and of cost at most B . Step 3 enumerates an entire small candidate set. Steps 4 and 5 are the two conclusions of [Theorem 4.5](#). Thus every terminal answer and every branch is justified by one of the explicit certificates above.

Although Step 4 is nonconstructive, a standard self-reduction recovers a witness: fix one label at a time by retaining the first tentative assignment whose child remains feasible. Thus no constructive implementation of the Wanless–Wood theorem is needed.

Lemma 4.6 (Correctness of the recursion). *Let $d \geq 2$, and consider the cheap-prefix decision procedure above on a $K_{d,d}$ -free graph. Every recursive state satisfying (I1)–(I3) returns yes if and only if there exists a tuple $(x_i)_{i \in L} \in \prod_{i \in L} A_i$ whose entries are distinct and pairwise nonadjacent and whose total cost is at most B .*

Proof. We induct on $r = |L|$. If $B < 0$, no tuple of nonnegative total cost is feasible; if some residual candidate set is empty, no completion exists. For $r = 0$, the empty residual tuple is feasible. For $r = 1$, every vertex in the nonempty residual candidate set is compatible with X_F and affordable by (I3). These are exactly the base cases in Step 1.

For soundness, a Step 4 yes-answer is valid by [Theorem 4.5\(a\)](#). In a branch, the induction hypothesis supplies a child completion avoiding $N[v]$; adding v yields a valid parent completion of cost at most $B' + c_i(v) = B$.

For completeness, fix a feasible completion $(x_i)_{i \in L}$. At a state with a small candidate set, Step 3 explicitly contains the branch $v = x_i$ for the chosen small label. If all prefixes are full and affordable, Step 4 returns yes. If they are full and over budget, [Theorem 4.5\(b\)](#) gives a label i with $x_i \in P_i$, and Step 5 includes the branch (i, x_i) . Independence places every other x_j outside $N[x_i]$, and nonnegative costs keep the remaining tuple within budget $B - c_i(x_i)$. Hence the rest of the fixed completion survives in the child, and induction applies. $\square$

4.6 Running time, witnesses, and optimization

Proof of Theorem 4.4. The root satisfies the invariants, so [Theorem 4.6](#) proves budget feasibility. Every branch assigns one label, and therefore the depth is at most k . The function $M(d, r)$ is nondecreasing in r . A branching state consequently has at most $rM(d, r) \leq kM(d, k)$ children. The search tree has $O((kM(d, k))^k)$ states. The three recursive state types and these depth and branching bounds are summarized in [Figure 1](#).

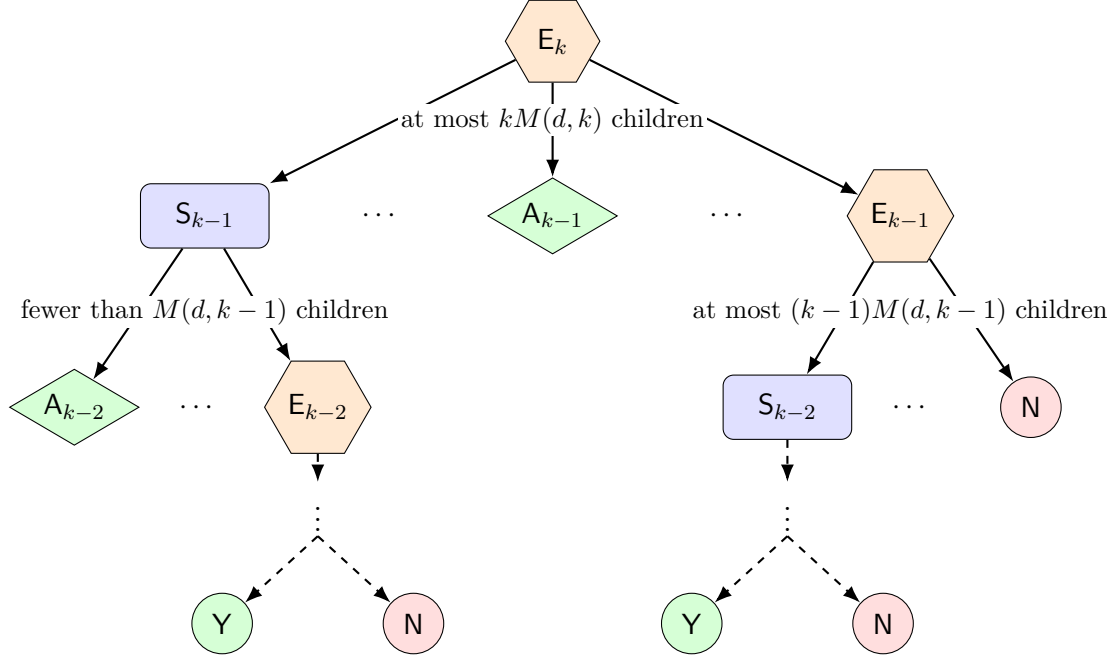


Figure 1: A schematic bounded search tree for the WEIGHTED INDEPENDENT TRANSVERSAL algorithm. A node S_r represents case (i): some residual candidate set A_i has size smaller than $M(d, r)$, and the algorithm branches on all vertices of A_i , producing fewer than $M(d, r)$ children. A node A_r represents case (ii): all cheap prefixes are full and collectively affordable. By Theorems 4.2 and 4.3, this is a terminal yes-node. A node E_r represents case (iii): all prefixes are full but collectively too expensive. By Theorem 4.5(b), every feasible completion meets some prefix, so the algorithm branches on at most $rM(d, r)$ label-vertex pairs. The symbols Y and N denote terminal yes- and no-states arising from the base cases. Every recursive edge fixes one label, so r decreases by one. Consequently, the tree has depth at most k , maximum branching factor $kM(d, k)$, and $O((kM(d, k))^k)$ nodes. The root is drawn as an E_k -node only to illustrate the largest branching case; in an actual instance it may instead be an S_k - or A_k -node, or terminate immediately in a base case.

At each state, filtering candidate sets, selecting prefixes, testing equality and adjacency, and performing exact rational arithmetic take polynomial time in $|Z|$. This proves the first bound in (6). By (5),

$$M(d, k) \leq (d-1)(4k)^d + 4d^2k,$$

and therefore

$$\log(kM(d, k)) = O(d \log k).$$

Raising the branching factor to depth k gives the second bound in (6). A depth-first implementation stores only one root-to-leaf assignment and its current candidate sets and prefixes.

To recover a witness, start at any state certified feasible by the decision procedure. Choose an unassigned label i and test its residual vertices one at a time. For each $v \in A_i$, form the child in (7) and call the decision procedure. Some child must be feasible; fix the first such v and continue. After at most k rounds, all labels are assigned. Across the rounds, at most $\sum_i |A_i^0|$ vertices are tested, so the number of decision calls is polynomial in $|Z|$ and is absorbed by (6).

For optimization, clear all rational denominators using a common denominator D , whose binary length is polynomial in the input size. If some original candidate set is empty, no

transversal exists. Otherwise, let

$$U = \sum_{i \in [k]} \max_{v \in A_i^0} Dc_i(v).$$

Every transversal has scaled cost at most U . We first run the decision procedure with budget U ; if it returns no, no transversal exists. Otherwise, binary search on the integer interval $[0, U]$ finds the least feasible scaled budget, and self-reduction recovers a transversal of that cost. Dividing by D gives the optimum in the original scale. The bit lengths of D and U and the number of decision calls are polynomial in $|I|$. $\square$

4.7 Completing the proof of the main theorem

Proof of Theorem 1.2. Apply Theorem 3.3 to the optimization algorithm of Theorem 4.4, with $p = d$ and $\mathcal{C}_d$ the class of $K_{d,d}$ -free graphs. The resulting WEIGHTED INDEPENDENT TRANSVERSAL instance has k candidate sets and polynomial encoding length. The transfer and Theorem 4.4 therefore compute $\text{OPT}(G, S)$ in time

$$2^{O(dk \log k)}(n + m)^{O(1)}.$$

If the optimum is finite, the transfer returns an optimal independent target and reconstructs a collision-free sequence of exactly that length; if no transversal exists, it reports that no independent k -set is reachable from S . For the budgeted problem, compare the computed optimum with b , incurring only polynomial dependence on the encoding length of b . $\square$

The cases $k \leq 1$ and $d = 1$ are immediate.

5 Further extensions

The proof in Section 4 has a graph-independent part and a graph-specific part. The graph-independent part is the cheap-prefix recursion: once we know a threshold such that every collection of sufficiently large candidate sets has compatible representatives, the three cases of the recursion and their cost argument are unchanged. The graph-specific part is only the estimate on conflicts between two candidate sets. We isolate this common calculation before deriving four sparsity consequences.

For possibly overlapping sets $U, V \subseteq V(G)$, let $G[U \rightarrow V]$ be the bipartite graph with disjoint parts

$$U^L = \{u^L : u \in U\}, \quad V^R = \{v^R : v \in V\},$$

where $u^L v^R$ is an edge exactly when $uv \in E(G)$. The arrow distinguishes the left and right copies; it does not orient G . If $|U| = |V| = \ell$, then $|E(G[U \rightarrow V])|$ is the number of ordered choices $(u, v) \in U \times V$ that conflict by adjacency. Exactly $|U \cap V|$ further ordered choices conflict by equality. Thus

$$\chi_G(U, V) = |E(G[U \rightarrow V])| + |U \cap V| \tag{8}$$

is the exact number of adjacency-or-equality conflicts among the ℓ^2 ordered choices.

Suppose that r candidate sets, each of size ℓ , are replaced by disjoint copied blocks as in the proof of Theorem 4.2. If $\chi_G(U, V) \leq \rho(\ell)\ell^2$ for every pair of blocks, then at most $(r-1)\rho(\ell)\ell^2$ conflict edges cross any one block. Therefore the Wanless–Wood criterion applies whenever

$$4(r-1)\rho(\ell) < 1. \tag{9}$$

The following lemma packages the universal analysis.

Lemma 5.1 (Sparse-prefix transfer principle). *Let $k \geq 2$, let G be a graph, and let $\mathcal{I}$ be a WEIGHTED INDEPENDENT TRANSVERSAL instance with k candidate sets on G . Suppose that a function $\gamma : \mathbb{N} \rightarrow \mathbb{R}_{\geq 0}$ satisfies*

$$|E(G[U \rightarrow V])| \leq \gamma(\ell)\ell^2$$

for every $\ell \in \mathbb{N}$ and all possibly overlapping $U, V \subseteq V(G)$ with $|U| = |V| = \ell$. Suppose the algorithm is also given, for every $r \in \{2, \dots, k\}$, a positive integer $L(r)$ satisfying

$$4(r-1) \left(\gamma(L(r)) + \frac{1}{L(r)} \right) < 1, \quad (10)$$

and put $\hat{L}(k) = \max_{2 \leq r \leq k} L(r)$. Then the decision version can be solved deterministically, with a feasible tuple returned for every yes-instance, in time

$$O\left((k\hat{L}(k))^k |\mathcal{I}|^{O(1)}\right).$$

Within the same bound, the optimization version returns a globally minimum-cost transversal or reports that none exists.

Proof. For two $L(r)$ -element candidate sets, adjacency contributes at most $\gamma(L(r))L(r)^2$ conflicts and equality contributes at most $L(r)$. Hence their total conflict proportion is at most $\gamma(L(r)) + 1/L(r)$. By (9) and (10), every r such sets have distinct, pairwise nonadjacent representatives.

Run the decision procedure of Section 4 with $L(r)$ in place of $M(d, r)$ when r labels remain. A small candidate set is still enumerated in full. If all prefixes are full and collectively affordable, the preceding certificate supplies a feasible completion. If they are collectively too expensive, the sorted-cost argument in Theorem 4.5(b) still shows that every feasible completion meets some prefix. Thus the correctness proof in Theorem 4.6 applies verbatim.

The recursion has depth at most k and branching factor at most $k\hat{L}(k)$. Each state takes polynomial time, giving the displayed bound. The self-reduction and rational-cost scaling in the proof of Theorem 4.4 give a witness and an optimum within the same parameterized bound. $\square$

5.1 Edge count: an edge-count-sensitive XP algorithm

The *edge count* of G is $m = |E(G)|$. This is a global measure: it does not prevent all m edges from being concentrated in a comparatively small vertex set. Nevertheless, it improves over enumerating all n^k destination tuples when $m \ll n^2$.

For $m \in \mathbb{N}_0$ and $r \geq 2$, define

$$L_e(m, r) = 8(r-1) + \left\lceil 4\sqrt{m(r-1)} \right\rceil + 1. \quad (11)$$

Theorem 5.2 (Edge-count-sensitive algorithm). *Let $k \geq 2$, and let $\mathcal{I}$ be a WEIGHTED INDEPENDENT TRANSVERSAL instance with k candidate sets whose underlying graph G has m edges. A minimum-cost transversal can be found deterministically, or nonexistence reported, in time*

$$2^{O(k \log k)} (m+1)^{k/2} |\mathcal{I}|^{O(1)}.$$

For INDEPENDENT SET DISCOVERY, given an n -vertex, m -edge graph G and a set $S \in \binom{V(G)}{k}$, one can compute $\text{OPT}(G, S)$ in time

$$2^{O(k \log k)} (m+1)^{k/2} (n+m)^{O(1)}.$$

If the optimum is finite, the algorithm also returns an optimal independent k -set and a shortest collision-free slide sequence; otherwise, it reports that no independent k -set is reachable from S . In particular, when $m = O(n)$ this is $2^{O(k \log k)} n^{k/2+O(1)}$ time.

Proof of Theorem 5.2. Fix $U, V \subseteq V(G)$ with $|U| = |V| = \ell$. An undirected edge xy of G contributes at most the two ordered adjacency pairs (x, y) and (y, x) to $U \times V$. Therefore

$$|E(G[U \rightarrow V])| \leq 2m, \quad \gamma_e(\ell) = \frac{2m}{\ell^2}.$$

For each $r \in \{2, \dots, k\}$, put $R = r - 1$ and $\ell = L_e(m, r)$. Then $\ell > 8R$ and $\ell > 4\sqrt{mR}$, so

$$4R \left(\frac{2m}{\ell^2} + \frac{1}{\ell} \right) < \frac{1}{2} + \frac{1}{2} = 1.$$

Thus Theorem 5.1 applies. The threshold is nondecreasing in r , and

$$L_e(m, k) = O(k + \sqrt{km}).$$

Consequently,

$$(kL_e(m, k))^k \leq 2^{O(k \log k)} (m+1)^{k/2},$$

which proves the WEIGHTED INDEPENDENT TRANSVERSAL bound. The INDEPENDENT SET DISCOVERY statement follows from the algorithmic transfer in Theorem 3.3. If $m = O(n)$, the last display is $2^{O(k \log k)} n^{k/2}$, as claimed. $\square$

5.2 A direct proof for bounded-degeneracy graphs

A graph is a -degenerate if every nonempty subgraph has a vertex of degree at most a . Equivalently, its vertices can be ordered so that every vertex has at most a neighbors later in the order. In particular, every h -vertex subgraph has at most ah edges.

For $a \in \mathbb{N}_0$ and $r \geq 2$, define

$$L_{\deg}(a, r) = 4(r-1)(4a+1) + 1. \quad (12)$$

Theorem 5.3 (Direct bounded-degeneracy algorithm). *Let $a \in \mathbb{N}_0$ and $k \geq 2$. Given a WEIGHTED INDEPENDENT TRANSVERSAL instance $\mathcal{I}$ with k candidate sets whose underlying graph G is a -degenerate, one can find a minimum-cost transversal, or report that none exists, in time*

$$2^{O(k \log k)} (a+1)^k |\mathcal{I}|^{O(1)}.$$

For INDEPENDENT SET DISCOVERY, given an n -vertex, m -edge, a -degenerate graph G and a set $S \in \binom{V(G)}{k}$, one can compute $\text{OPT}(G, S)$ in time

$$2^{O(k \log k)} (a+1)^k (n+m)^{O(1)}.$$

If the optimum is finite, the algorithm also returns an optimal independent k -set and a shortest collision-free slide sequence; otherwise, it reports that no independent k -set is reachable from S . Consequently, both problems are uniformly FPT in $k + a$.

Proof of Theorem 5.3. Let $U, V \subseteq V(G)$ have size ℓ and put $W = U \cup V$. Degeneracy gives $|E(G[W])| \leq a|W| \leq 2a\ell$. Each edge of $G[W]$ contributes at most two ordered pairs to $E(G[U \rightarrow V])$, and hence

$$|E(G[U \rightarrow V])| \leq 4a\ell, \quad \gamma_{\deg}(\ell) = \frac{4a}{\ell}.$$

For each $r \in \{2, \dots, k\}$, set $\ell = L_{\deg}(a, r)$. Then

$$4(r-1) \left(\gamma_{\deg}(\ell) + \frac{1}{\ell} \right) = \frac{4(r-1)(4a+1)}{\ell} < 1.$$

Apply [Theorem 5.1](#). Since $L_{\deg}(a, k) = O((a+1)k)$, its running time becomes

$$O\left((kL_{\deg}(a, k))^k |\mathcal{I}|^{O(1)}\right) = 2^{O(k \log k)} (a+1)^k |\mathcal{I}|^{O(1)}.$$

The INDEPENDENT SET DISCOVERY clause again follows from [Theorem 3.3](#). A graph's degeneracy and a corresponding ordering can be computed in $O(n+m)$ time, so the value a need not be supplied separately. $\square$

Fellows et al. [6] previously established FPT in k on every fixed bounded-degeneracy class using independence-covering families. The argument above is a short and structurally different proof: one local edge count feeds directly into the cheap-prefix recursion, handles coordinate-dependent costs and overlapping candidate sets, and makes the dependence on a explicit. It is also quantitatively sharper than applying our biclique-free theorem as a black box. For $a \geq 1$, an a -degenerate graph is $K_{a+1, a+1}$ -free, but substituting $d = a+1$ in [Theorem 1.2](#) gives the factor $2^{O((a+1)k \log k)}$, whereas the direct analysis gives $2^{O(k \log k + k \log(a+1))}$.

5.3 Bounded s -codegree

For an integer $s \geq 1$, define the s -codegree of a graph G by

$$\lambda_s(G) = \max_{X \in \binom{V(G)}{s}} \left| \bigcap_{x \in X} N_G(x) \right|, \quad (13)$$

with $\lambda_s(G) = 0$ when $|V(G)| < s$. Thus $\lambda_1(G)$ is the maximum degree $\Delta(G)$, and $\lambda_2(G)$ is the usual maximum codegree, the largest number of common neighbors of a pair of vertices.

Fix $q \in \mathbb{N}_0$ and assume that $\lambda_s(G) \leq q$. The asymmetric Kővári–Sós–Turán bound [10] says that a bipartite graph with ℓ vertices in each part and with no $K_{s, q+1}$ having its s -vertex side in the left part has at most

$$q^{1/s} \ell^{2-1/s} + (s-1)\ell \quad (14)$$

edges. For any ℓ -element sets $U, V \subseteq V(G)$, the copied graph $G[U \rightarrow V]$ satisfies this hypothesis. Indeed, the s left vertices of a copied $K_{s, q+1}$ would project to s distinct vertices of G with at least $q+1$ common neighbors. No projected vertex can occur on both sides, because the corresponding copied cross-edge would require a loop in G .

For $r \geq 2$, define

$$L_{s, q}(r) = q(8(r-1))^s + 8s(r-1) + 1. \quad (15)$$

Theorem 5.4 (Bounded s -codegree). *Let $k \geq 2$, $s \geq 1$, and $q \geq 0$ be integers. Given s, q and a WEIGHTED INDEPENDENT TRANSVERSAL instance $\mathcal{I}$ with k candidate sets whose underlying graph G is promised to satisfy $\lambda_s(G) \leq q$, one can find a minimum-cost transversal, or report that none exists, in time*

$$2^{O(sk \log k + k \log(q+1))} |\mathcal{I}|^{O(1)}.$$

For INDEPENDENT SET DISCOVERY, given s, q , an n -vertex, m -edge graph G promised to satisfy $\lambda_s(G) \leq q$, and a set $S \in \binom{V(G)}{k}$, one can compute $\text{OPT}(G, S)$ in time

$$2^{O(sk \log k + k \log(q+1))} (n+m)^{O(1)}.$$

If the optimum is finite, the algorithm also returns an optimal independent k -set and a shortest collision-free slide sequence; otherwise, it reports that no independent k -set is reachable from S . Thus the problems are FPT in $k+s+q$ and, for fixed s , in $k+q$.

The promise $\lambda_s(G) \leq q$ need not be verified.

Proof of Theorem 5.4. Dividing (14) by ℓ^2 gives the adjacency-density bound

$$\gamma_{s,q}(\ell) = q^{1/s} \ell^{-1/s} + \frac{s-1}{\ell}.$$

After equality conflicts are included, the expression in (10) is

$$q^{1/s} \ell^{-1/s} + \frac{s}{\ell}.$$

For each $r \in \{2, \dots, k\}$, put $R = r - 1$ and $\ell = L_{s,q}(r)$. If $q > 0$, then $\ell > q(8R)^s$ and hence $q^{1/s} \ell^{-1/s} < 1/(8R)$; for $q = 0$, this term is zero. Also $\ell > 8sR$, so $s/\ell < 1/(8R)$. It follows that

$$4R \left(q^{1/s} \ell^{-1/s} + \frac{s}{\ell} \right) < 1.$$

The sparse-prefix transfer principle applies. Since the threshold is nondecreasing in r and

$$L_{s,q}(k) \leq q(8k)^s + 8sk + 1,$$

we obtain

$$(kL_{s,q}(k))^k = 2^{O(sk \log k + k \log(q+1))}.$$

This proves the static bound, and Theorem 3.3 gives the INDEPENDENT SET DISCOVERY bound. $\square$

The case $s = 1$ deserves emphasis: (13) becomes $\lambda_1(G) = \Delta(G)$, and (15) is $O((q+1)r)$. Thus bounded maximum degree is exactly the first member of the s -codegree hierarchy, not a separate case. For $s = 2$ and $q \geq 1$, the threshold is $O(qr^2 + r)$, which can be substantially smaller than treating the graph only through the balanced biclique $K_{q+1,q+1}$ that it excludes.

5.4 $K_{s,t}$ -free graphs

For integers $1 \leq s \leq t$, the complete bipartite graph $K_{s,t}$ has one part of size s , one part of size t , and every possible edge between the two parts. A graph is $K_{s,t}$ -free if it contains no such graph as a subgraph; the forbidden copy need not be induced. By the definition of a common neighborhood,

$$G \text{ is } K_{s,t}\text{-free} \iff \lambda_s(G) \leq t - 1. \quad (16)$$

Corollary 5.5 (Unbalanced biclique exclusion). *Let $k \geq 2$ and let $1 \leq s \leq t$ be integers. Given a WEIGHTED INDEPENDENT TRANSVERSAL instance $\mathcal{I}$ with k candidate sets whose underlying graph G is promised to be $K_{s,t}$ -free, one can find a minimum-cost transversal, or report that none exists, in time*

$$2^{O(sk \log k + k \log t)} |\mathcal{I}|^{O(1)}.$$

For INDEPENDENT SET DISCOVERY, given an n -vertex, m -edge graph G promised to be $K_{s,t}$ -free and a set $S \in \binom{V(G)}{k}$, one can compute $\text{OPT}(G, S)$ in time

$$2^{O(sk \log k + k \log t)} (n + m)^{O(1)}.$$

If the optimum is finite, the algorithm also returns an optimal independent k -set and a shortest collision-free slide sequence; otherwise, it reports that no independent k -set is reachable from S . In particular, both problems are FPT in $k + s + t$ and in k for every fixed pair (s, t) .

Proof of Theorem 5.5. Apply Theorem 5.4 with $q = t - 1$. Equivalently, use the explicit threshold, for every $r \in \{2, \dots, k\}$,

$$L_{s,t}^{\text{bic}}(r) = (t-1)(8(r-1))^s + 8s(r-1) + 1.$$

Its logarithm is $O(s \log k + \log t)$ for $r \leq k$, so the search-tree factor is $2^{O(sk \log k + k \log t)}$. The equivalence in (16) proves that the promise needed by Theorem 5.4 holds. $\square$

For $t \geq 2$, the direct asymmetric analysis strengthens the bound obtained by treating a $K_{s,t}$ -free graph as merely $K_{t,t}$ -free. The latter observation already gives FPT in k for fixed s, t , but it places t in the extremal exponent; Theorem 5.5 instead puts the smaller side s in that exponent, while the larger side enters only linearly in the prefix threshold.

5.5 Weighted movement on a separate graph

The proof uses biclique exclusion only for terminal conflicts. The graph in which tokens move may be different.

Corollary 5.6 (Two-graph, weighted movement). *Let $d, k \geq 2$ be integers. Let G_f be an undirected graph promised to be $K_{d,d}$ -free, and let G_m be an arbitrary directed graph on a common n -vertex set V . Let $w : E(G_m) \rightarrow \mathbb{Q}_{\geq 0}$ be a binary-encoded arc-cost function shared by all tokens, and let $S = \{s_1, \dots, s_k\} \subseteq V$. A move sends one token along an arc of G_m to an unoccupied vertex, and only the terminal configuration must be independent in G_f . The total movement cost of a sequence is the sum, with multiplicity, of the costs of its traversed arcs. Given $\mathcal{J} = (d, G_f, G_m, w, S)$, one can deterministically find a G_f -independent terminal k -set and a collision-free sequence that lexicographically minimizes*

$$(\text{total movement cost, number of slides})$$

over all such sequences, in time $2^{O(dk \log k)} |\mathcal{J}|^{O(1)}$. If no such terminal configuration is reachable, the algorithm reports nonexistence. Here $|\mathcal{J}|$ is the total binary encoding length.

Proof. Multiply all arc costs by the product of their denominators. This produces nonnegative integer costs with polynomial encoding length and does not change the lexicographic optimum; henceforth w denotes these scaled integer costs. Give an arc e the two-component length $\ell(e) = (w(e), 1)$, where the first component records cost and the second records the number of slides. Vectors are added componentwise and compared lexicographically; $+\infty$ is larger than every finite vector and is absorbing under addition. For $u, v \in V$, let $\delta(u, v) \in \mathbb{N}_0^2 \cup \{+\infty\}$ be the lexicographically minimum length of a directed u - v path, or $+\infty$ if no such path exists. For two k -sets Q, T , define

$$\mu^{\text{lex}}(Q, T) = \min_{\pi: Q \rightarrow T \text{ bijective}} \sum_{u \in Q} \delta(u, \pi(u)).$$

We first show that this assignment value equals the lexicographically minimum cost-slide pair of a collision-free sequence. Every slide sequence induces a source-target bijection and therefore has value at least $\mu^{\text{lex}}(Q, T)$. Thus no sequence exists when $\mu^{\text{lex}}(Q, T) = +\infty$. Suppose henceforth that $\mu^{\text{lex}}(Q, T) < +\infty$. If $Q = T$, the empty sequence attains this value. Otherwise, fix an optimal bijection π , choose $t \in T \setminus Q$, and put $s = \pi^{-1}(t)$. Follow a lexicographically shortest directed s - t path until its first unoccupied vertex v , and let u be the occupied predecessor of v .

If $u \neq s$, put $t_u = \pi(u)$ and exchange the destinations of the tokens at s and u . The directed triangle inequality and the shortest-path equality $\delta(s, t) = \delta(s, u) + \delta(u, t)$ give

$$\begin{aligned}\delta(u, t) + \delta(s, t_u) &\leq \delta(u, t) + \delta(s, u) + \delta(u, t_u) \\ &= \delta(s, t) + \delta(u, t_u).\end{aligned}$$

Thus the exchange cannot increase the assignment value. Since the original assignment was optimal, the exchanged one is also optimal and assigns the token at u to t . If $u = s$, the same conclusion holds without an exchange.

Now slide the token from u to v , obtaining $Q' = (Q \setminus \{u\}) \cup \{v\}$. Because the u - t path begins with the arc (u, v) and its suffix is shortest, replacing source u by v in the current assignment gives

$$\ell(u, v) + \mu^{\text{lex}}(Q', T) \leq \mu^{\text{lex}}(Q, T).$$

Conversely, in any assignment from Q' to T , replace source v by u and keep its destination. The directed triangle inequality gives

$$\mu^{\text{lex}}(Q, T) \leq \ell(u, v) + \mu^{\text{lex}}(Q', T).$$

Hence $\mu^{\text{lex}}(Q, T) = \ell(u, v) + \mu^{\text{lex}}(Q', T)$. Lexicographic order on $\mathbb{N}_0^2$ is well-founded, and the second component decreases even when $w(u, v) = 0$. Repeating the argument therefore constructs a collision-free sequence whose cost-slide pair is exactly $\mu^{\text{lex}}(Q, T)$.

Every lexicographically shortest path is simple: deleting a directed cycle does not increase its cost and strictly decreases its number of arcs. Thus each assigned path uses at most $n - 1$ arcs, and the total secondary component is at most $k(n - 1)$. Set $C = k(n - 1) + 1$. For every arc, the scalar length

$$Cw(e) + 1$$

preserves the lexicographic order exactly. Indeed, for any collection of k simple directed paths, the total number b of arcs lies in $[0, C - 1]$. If (a, b) and (a', b') are the cost-arc vectors of two such collections, then $Ca + b < Ca' + b'$ if and only if (a, b) is lexicographically smaller than (a', b') : when $a < a'$, integrality gives $Ca + b \leq Ca + C - 1 < C(a + 1) \leq Ca' + b'$, and when $a = a'$, the scalar comparison is exactly the comparison of b and b' . A shortest-path computation from each source therefore supplies the finite candidate set A_i of vertices reachable from s_i and, for every $v \in A_i$, the coordinate cost

$$c_i(v) = \min_{P: s_i \rightsquigarrow v} \sum_{e \in E(P)} (Cw(e) + 1),$$

where the minimum ranges over directed s_i - v paths in G_m .

Apply [Theorem 4.4](#) to these candidate sets in the feasibility graph G_f . If no transversal exists, then no reachable independent terminal configuration exists. Otherwise, the selected destinations are distinct and independent in G_f . For the resulting target set T , a lexicographically cheaper source-destination bijection would itself define a cheaper feasible WEIGHTED INDEPENDENT TRANSVERSAL tuple, contradicting optimality. Hence the selected pairing attains $\mu^{\text{lex}}(S, T)$, and the directed first-unoccupied-vertex construction above recovers an optimal collision-free sequence. The scaling, shortest paths, and reconstruction all have polynomial bit complexity and polynomial running time outside the call to [Theorem 4.4](#), which proves the stated bound. $\square$

References

- [1] D. Bilò, L. Gualà, S. Leucci, and G. Proietti. Exact and approximate algorithms for movement problems on (special classes of) graphs. *Theoretical Computer Science*, 652:86–101, 2016. DOI: [10.1016/j.tcs.2016.09.007](https://doi.org/10.1016/j.tcs.2016.09.007).

- [2] N. Bousquet, A. E. Mouawad, S. Maaz, N. Nishimura, and S. Siebertz. On algorithmic meta-theorems for solution discovery: Tractability and barriers. *arXiv preprint arXiv:2510.17344*, 2025. DOI: [10.48550/arXiv.2510.17344](https://doi.org/10.48550/arXiv.2510.17344).
- [3] G. Călinescu, A. Dumitrescu, and J. Pach. Reconfigurations in graphs and grids. *SIAM Journal on Discrete Mathematics*, 22(1):124–138, 2008. DOI: [10.1137/060652063](https://doi.org/10.1137/060652063).
- [4] E. D. Demaine, M. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, and M. Zadimoghaddam. Minimizing movement. *ACM Transactions on Algorithms*, 5(3):30:1–30:30, 2009. DOI: [10.1145/1541885.1541891](https://doi.org/10.1145/1541885.1541891).
- [5] E. D. Demaine, M. Hajiaghayi, and D. Marx. Minimizing movement: Fixed-parameter tractability. *ACM Transactions on Algorithms*, 11(2):14:1–14:29, 2014. DOI: [10.1145/2650247](https://doi.org/10.1145/2650247).
- [6] M. R. Fellows, M. Grobler, N. Megow, A. E. Mouawad, V. Ramamoorthi, F. A. Rosamond, D. Schmand, and S. Siebertz. On solution discovery via reconfiguration. *Journal of Computer and System Sciences*, 157:103747, 2026. DOI: [10.1016/j.jcss.2025.103747](https://doi.org/10.1016/j.jcss.2025.103747).
- [7] E. Gerhard, S. Maaz, P. Schott, S. Siebertz, and J. Wodtke. FO value discovery and partial vertex cover discovery. *arXiv preprint arXiv:2607.06446*, 2026. DOI: [10.48550/arXiv.2607.06446](https://doi.org/10.48550/arXiv.2607.06446).
- [8] M. Grobler, S. Maaz, N. Megow, A. E. Mouawad, V. Ramamoorthi, D. Schmand, and S. Siebertz. Solution discovery via reconfiguration for problems in P. In *51st International Colloquium on Automata, Languages, and Programming (ICALP 2024)*, volume 297 of *LIPIcs*, pages 76:1–76:20, 2024. DOI: [10.4230/LIPIcs.ICALP.2024.76](https://doi.org/10.4230/LIPIcs.ICALP.2024.76).
- [9] M. Grobler, S. Maaz, A. E. Mouawad, N. Nishimura, V. Ramamoorthi, and S. Siebertz. Kernelization complexity of solution discovery problems. In *35th International Symposium on Algorithms and Computation (ISAAC 2024)*, volume 322 of *LIPIcs*, pages 36:1–36:17, 2024. DOI: [10.4230/LIPIcs.ISAAC.2024.36](https://doi.org/10.4230/LIPIcs.ISAAC.2024.36).
- [10] T. Kővári, V. T. Sós, and P. Turán. On a problem of K. Zarankiewicz. *Colloquium Mathematicum*, 3(1):50–57, 1954. DOI: [10.4064/cm-3-1-50-57](https://doi.org/10.4064/cm-3-1-50-57).
- [11] M. Kuhn, D. Lokshtanov, and Z. Miller. Lower bound for independence covering in C_4 -free graphs. *ACM Transactions on Computation Theory*, 16(3):16:1–16:7, 2024. DOI: [10.1145/3664277](https://doi.org/10.1145/3664277).
- [12] D. Lokshtanov, F. Panolan, S. Saurabh, R. Sharma, and M. Zehavi. Covering small independent sets and separators with applications to parameterized algorithms. *ACM Transactions on Algorithms*, 16(3):32:1–32:31, 2020. DOI: [10.1145/3379698](https://doi.org/10.1145/3379698).
- [13] R. Saito, A. Sommer, T. Suga, T. Suzuki, and Y. Tamura. Solution discovery for vertex cover, independent set, dominating set, and feedback vertex set. In *SOFSEM 2026: Theory and Practice of Computer Science*, volume 16448 of *Lecture Notes in Computer Science*, pages 432–446. Springer, 2026. DOI: [10.1007/978-3-032-17801-5_32](https://doi.org/10.1007/978-3-032-17801-5_32).
- [14] H. von Bergen, L. Fastenau, E. Gerhard, N. Lorenz, S. Maaz, A. E. Mouawad, R. Rabinovich, N. Schirmacher, D. Schmand, S. Siebertz, and M. Trinh. Separating feasibility and movement in solution discovery: The case of path discovery. *arXiv preprint arXiv:2604.27802*, 2026. DOI: [10.48550/arXiv.2604.27802](https://doi.org/10.48550/arXiv.2604.27802).
- [15] I. M. Wanless and D. R. Wood. A general framework for hypergraph coloring. *SIAM Journal on Discrete Mathematics*, 36(3):1663–1677, 2022. DOI: [10.1137/21M1421015](https://doi.org/10.1137/21M1421015).