QPTAS for Max Weight Independent Set in Graphs with Strongly Edge-Sublinear Separators

Édouard Bonnet
CNRS, ENS de Lyon, Université Claude Bernard Lyon 1, LIP UMR 5668, Lyon, France

Yeonsu Chang
Department of Mathematics, Hanyang University, Seoul, South Korea

Abstract

A hereditary graph class has strongly edge-sublinear separators if, for some constant $0 \leqslant b < 1$, its m-edge graphs admit balanced separators of size $\mathcal{O}(m^b)$. We show that Max Weight Independent Set admits a deterministic quasipolynomial-time approximation scheme (QPTAS) on every such class. In particular, this holds on graphs excluding any fixed graph as an induced minor, by a separator theorem of [Korhonen and Lokshtanov, SODA '24].

Our theorem generalizes the QPTASes for axis-parallel rectangles [Adamaszek and Wiese, FOCS '13] and polygons [Adamaszek, Har-Peled, and Wiese, JACM '19], or equivalently for packing connected subgraphs of a planar graph [Pilipczuk, van Leeuwen, and Wiese, Algorithmica '20]. It also generalizes the QPTAS for MWIS on hereditary classes with strongly sublinear separators measured in vertices [Dvořák, EJC '18; Dvořák and Lahiri, ESA '21], as well as the QPTASes of [Chudnovsky et al., arXiv '25], [Bonnet et al., SWAT '26], and [Czyżewska et al., arXiv '26] for graphs excluding specific induced minors.

1 Introduction

Max Independent Set (MIS) asks for a largest set of pairwise nonadjacent vertices in a graph G. This problem belongs to Karp's list of NP-hard problems [38]. Furthermore, Håstad's hardness result [36], derandomized by Zuckerman [49], shows that, for every fixed $\varepsilon > 0$, approximating MIS within a factor $n^{1-\varepsilon}$ on n-vertex graphs is NP-hard. This makes MIS a particularly inapproximable problem.

However, in some special classes of graphs, MIS becomes (much more) tractable. For instance, it admits a polynomial-time exact algorithm on graphs of bounded treewidth [3], graphs of bounded clique-width [17, 44], P_6-free graphs [33], and circle graphs [29]; a polynomial-time approximation scheme (PTAS) on planar graphs [4] and, more generally, classes excluding a fixed minor [30], on pseudo-disk intersection graphs [13], and on represented disk graphs with arbitrary radii [25]; an exact quasipolynomial-time algorithm on P_t-free graphs for every fixed t [27] and on graphs with no induced cycle of length at least t for every fixed $t \geqslant 4$ [28]; and a quasipolynomial-time approximation scheme (QPTAS) on represented intersection graphs of axis-parallel rectangles [2], polygons [1], and equivalently connected subgraphs of a planar graph [45].

Four years ago, Dallard, Milanič, and Štorgel [19, Question 8.2] asked if MIS admits a polynomial algorithm on every class excluding a fixed planar graph as an induced minor (i.e., a graph obtained via vertex deletions and edge contractions). A positive solution to this question would have been a formidable breakthrough: it would have unified, extended, and improved the various algorithms developed on the classes above with a known polynomial or quasipolynomial algorithm (indeed, these classes exclude some planar induced minors). Independently, Gartland and Lokshtanov conjectured the same [26, Conjecture 1.4.2], or at least that a quasipolynomial-time algorithm exists.

Recently, the authors [7] negatively resolved Question 8.2 of [19] (and refuted Gartland and Lokshtanov's conjecture) by showing that MIS remains NP-complete in graphs excluding the 5×5 grid as an induced minor; ruling out, unless NP $\subseteq$ QP, even a quasipolynomial

algorithm in this special case. This reduction, however, does not rule out approximation schemes.

The existence of a QPTAS is discussed in [26, Section 1.4]. A proposed route is through the so-called *structural induced grid minor conjecture* of Gartland and Lokshtanov [26, Conjecture 1.4.1]. This is the conjecture that, for every planar graph H (or equivalently every $k \times k$ grid), every graph excluding H as an induced minor has a balanced separator in the neighborhood of $\mathcal{O}_H(1)$ vertices. It is indeed known that proving this conjecture for some graph H yields a QPTAS for MIS on H-induced-minor-free graphs [26, Theorem 1.2.6].

▶ **Conjecture 1** (Gartland–Lokshtanov). *For every planar graph H,* Max Independent Set *admits a QPTAS in H-induced-minor-free graphs.*

This conjecture has been repeated in several papers, for example [7, Conjecture 7] or [43], as a challenging open question.

Our result. We positively resolve Conjecture 1 in much greater generality. We say that a hereditary graph class $\mathcal{C}$ has *strongly edge-sublinear separators* (or equivalently, *strongly edge-sublinear treewidth*) if there are constants $0 \leqslant b < 1$, $c > 0$ (only depending on $\mathcal{C}$) such that for every $m \geqslant 1$, every m-edge graph of $\mathcal{C}$ has a balanced separator of size at most $c \cdot m^b$.

Our algorithm also applies to Max Weight Independent Set (MWIS), where vertices have rational weights and the objective is to maximize the total weight of an independent set. We show the following theorem.

▶ **Theorem 2.** *Let $\mathcal{C}$ have strongly edge-sublinear separators. Then* Max Weight Independent Set *admits a QPTAS on $\mathcal{C}$.*

For $G \in \mathcal{C}$ and rational $0 < \varepsilon < 1$, the algorithm deterministically computes an independent set I of G with weight at least $(1-\varepsilon)\alpha_w(G)$ in time $\exp\big((\varepsilon^{-1}\log|V(G)|)^{\mathcal{O}_{\mathcal{C}}(1)}\big) + \ell^{\mathcal{O}(1)}$, where $\alpha_w(G)$ is the maximum weight of an independent set of G and ℓ is the total binary encoding length of the vertex weights and ε. No separator or algorithm for finding separators in $\mathcal{C}$ is required as part of the input.

The QPTAS is obtained through a different route than [26, Conjecture 1.4.1] (which anyway cannot hold in general H-induced-minor-free classes, let alone in classes with strongly edge-sublinear separators), so the structural induced grid minor conjecture remains open. By a separator theorem of Korhonen and Lokshtanov [39], any class excluding a fixed graph H (not necessarily planar) as an induced minor has strongly edge-sublinear separators with $b = \frac{1}{2} + \varepsilon$ for any $\varepsilon > 0$. Therefore, Theorem 2 implies the following corollary.

▶ **Corollary 3.** *For every fixed graph H,* Max Weight Independent Set *admits a QPTAS on H-induced-minor-free graphs.*

This unifies the QPTAS guarantee for all the classes listed in the second paragraph, including those admitting faster approximation schemes or exact algorithms, as each excludes a fixed graph as an induced minor. We note that having strongly edge-sublinear separators is more general a property than excluding an induced minor. For instance, all induced subgraphs of subdivisions of n-vertex graphs with at least $n^{0.1}$ subdivision vertices replacing each edge of the original graph is a separating example.

The weighted Nemhauser–Trotter reduction converts a $(1-\varepsilon)$-approximation for MWIS on a graph class into a $(1+\varepsilon)$-approximation for Minimum Weight Vertex Cover on the same class [34, Theorem 3.1]. Thus Theorem 2 also gives a deterministic QPTAS for this problem on every class with strongly edge-sublinear separators.

The QPTAS guarantee does not extend to all graph optimization problems. Minimum Dominating Set admits no QPTAS even on P_5-free graphs unless NP $\subseteq$ QP. This follows from the inapproximability of Set Cover [20] via the standard approximation-preserving reduction to split graphs, which are P_5-free. Under the same assumption, Maximum Clique admits no QPTAS on $2K_2$(-induced-minor)-free graphs, by complementation from the APX-hardness of MIS on C_4-free graphs [14, Theorem 3.6(i)].

Bonnet and Hickingbotham [9] posed the stronger conjecture that MIS admits a PTAS when excluding any graph as an induced minor. This remains open.

▶ **Conjecture 4** ([9]). *For every graph H,* Max Independent Set *admits a PTAS in H-induced-minor-free graphs.*

A natural strengthening is to ask Conjecture 4 for classes with strongly edge-sublinear separators, and a natural weakening is to ask the question for planar graphs H only. Conjecture 4 is even open in the particular case of represented intersection graphs of axis-parallel rectangles.

Related work. On classes with strongly sublinear separators (measured in vertices), MIS admits a PTAS [35] and MWIS a QPTAS [22, 23]. Theorem 2 generalizes the latter to separators measured in edges. A PTAS for MWIS is also known on *efficiently fractionally treewidth-fragile* classes [21, 23], i.e., where for every fixed $\varepsilon > 0$, one can compute in polynomial time a distribution on vertex sets X such that $\Pr[v \in X] \leqslant \varepsilon$ for every vertex v and each graph $G - X$ has treewidth bounded in terms of ε and the class.

String graphs, the intersection graphs of curves in the plane, have strongly edge-sublinear separators [42]. More generally, Lee [40] proves such separator bounds for *region intersection graphs* over K_t-minor-free hosts, for every fixed t. These are intersection graphs of connected subgraphs of the host graphs. Theorem 2 therefore gives a deterministic QPTAS for MWIS on these classes using only their adjacency relation. It recovers the schemes for represented intersection graphs of axis-parallel rectangles [2] and polygons [1], and for packing connected subgraphs of a planar graph [45], without requiring the corresponding representations.

Earlier QPTASes for MWIS under induced-minor exclusion cover C_t for $t \geqslant 4$ [16, Theorem 1.4], W_ℓ for $\ell \geqslant 3$ [15, Corollary 1.3], sC_t for $s \geqslant 1$ and $t \geqslant 3$ [8], and $k\Theta_t$ for $k, t \geqslant 1$ [18, Theorem 1.11], with all parameters fixed. Here W_ℓ is a cycle of length ℓ with an additional vertex adjacent to every cycle vertex, and Θ_t consists of three internally vertex-disjoint paths with common endpoints, each having t internal vertices. The notation sC_t and $k\Theta_t$ denotes disjoint unions of s cycles and k copies of Θ_t, respectively. All these results are special cases of Corollary 3. Nevertheless, some of these papers [15, 18] obtain their QPTAS by showing the still-open structural induced grid minor conjecture (for some planar graphs rather than grids).

Proof outline. We first deal with unweighted MIS. The algorithm finds a balanced separator of small weight with respect to some vertex costs. We delete the balanced separator, recursively find independent sets in the remaining components, and take their union. The separator choice and the vertex costs are informed by a linear programming relaxation. The algorithm either finds a suitable approximation or yields a valid inequality that is added to the linear program. This inequality is given to the ellipsoid algorithm. This technique is called round-or-cut. We now detail every step. The reader may want to go through Algorithms 1 and 2 before or after reading the more detailed outline, which follows.

Section 3 presents an algorithm that finds a balanced separator of G of small total cost when the vertex costs $x \in [0,1]^{V(G)}$ satisfy some conditions. (For $X \subseteq V(G)$, we set $x(X) :=$

$\sum_{v\in X} x_v$.) More precisely, for any small positive δ, there is a threshold $p = (\delta^{-1}\log n)^{\mathcal{O}(1)}$, such that

$$x(V(G)) = k \geqslant p \qquad \text{and} \qquad \sum_{uv\in E(G)} x_u x_v \leqslant k^2/p \tag{1}$$

guarantee a balanced separator S with $x(S) \leqslant \delta k$. The proof uses a vertex-capacitated version of the Leighton–Rao flow–cut theorem [41]. If the separator is not found, the resulting flow lets us represent the edges of a subcubic graph of large treewidth by sampling paths; this follows Korhonen and Lokshtanov [39]. The assumptions on x control the lengths, intersections, and adjacencies of these paths. The short paths induce a graph violating the assumption that the class has strongly edge-sublinear treewidth.

For a target $k > p$ for the size of the independent set, Section 4 introduces a type of Sherali–Adams relaxation [48] with variables y_A indexed by polylogarithmically small vertex sets; see (13). For every independent set I of size k, the assignment $y_A = \mathbf{1}[A \subseteq I]$ satisfies its constraints. For each k, we apply the ellipsoid algorithm to the convex hull of the encodings of independent sets of size k, using the original relaxation constraints and further inequalities obtained from recursive calls.

If the query y satisfies the constraints of the relaxation, an entropy-decrease argument shows that conditioning on a set A with $|A| = \mathcal{O}(p\log n)$ and $y_A > 0$ gives costs $x_v := y_{A\cup\{v\}}/y_A$ satisfying (1). (Conditioning on A fixes $x_v = 1$ for every $v \in A$.) Adding a vertex to A requires recomputing all these ratios, so the costs outside A change, too.

The entropy of the distribution $(x_v/k)_{v\in V(G)}$ lies between 0 and $\log n$. Whenever $\sum_{uv\in E(G)} x_u x_v > k^2/p$, conditioning on a suitable vertex decreases the entropy by more than $2/p$, which upper-bounds the number of vertices added to A. We find A deterministically by enumerating all sets up to this size, which is polylogarithmic with our choice of parameters. Targets $k \leqslant p$ are handled by exhaustive search.

In conclusion, Section 4 gives us vertex costs x_v satisfying (1), and with Section 3, we obtain a balanced separator S such that $x(S) \leqslant \delta k$ from a positive query. In Section 5, we use this separator S to recursively compute an independent set I_C and an upper bound $u_C \geqslant \alpha(G[C])$ for each component C of $G - S$. Let ρ be a common approximation factor guaranteed by the child calls, so that $|I_C| \geqslant \rho u_C$ for every C. Note that $I := \bigcup_C I_C$ is an independent set because distinct components have no edges between them. If $x(C) \leqslant u_C$ for every component C, then

$$|I| = \sum_C |I_C| \geqslant \rho \sum_C u_C \geqslant \rho \sum_C x(C) = \rho(k - x(S)) \geqslant \rho(1-\delta)k.$$

Each component has at most two thirds as many vertices as its parent, so there are at most $\lceil \log_{3/2} n \rceil$ separator steps along any branch of the recursion tree. Starting from exact solutions at the leaves, these factors multiply to give an independent set of size at least $(1-\delta)^{\lceil \log_{3/2} n \rceil} k$.

Otherwise, a component with $x(C) > u_C$ yields the inequality $\sum_{v\in C} z_{A\cup\{v\}} \leqslant u_C z_A$ in variables z with the same coordinates as y. This inequality holds for every vector z encoding an independent set of size k but is violated by the current query y.

We pass this inequality to the ellipsoid algorithm, which generates another query or certifies infeasibility. This is the round-or-cut approach of Carr et al. [11]. The procedure eventually constructs a sufficiently large independent set or proves $\alpha(G) < k$. Trying targets in decreasing order establishes the upper bounds required by the recursion: all targets above

the first successful one have been ruled out. Choosing $\delta := \varepsilon/\lceil \log_{3/2} n \rceil$ bounds the total approximation loss by ε, since Bernoulli's inequality gives

$$(1-\delta)^{\lceil \log_{3/2} n \rceil} \geqslant 1 - \delta \lceil \log_{3/2} n \rceil = 1 - \varepsilon.$$

Section 5.3 shows that the total running time is quasipolynomial.

Section 6 reduces MWIS to MIS by scaling and rounding the weights to integers, then replacing each vertex by that many pairwise nonadjacent copies. We prove that this operation preserves strongly edge-sublinear separators. Applying the unweighted QPTAS to the resulting graph and returning the original vertices with a selected copy gives the required approximation for MWIS.

2 Preliminaries

We write log for the base-2 logarithm. Subscripts in asymptotic notation indicate which parameters the implicit constants may depend on. For a statement P, the indicator $\mathbf{1}[P]$ equals 1 if P holds and 0 otherwise.

All graphs are finite, simple, and undirected unless explicitly stated otherwise. For a graph G, we write $V(G)$ and $E(G)$ for its vertex and edge sets. We denote by $\alpha(G)$ the *independence number* of G, i.e., the size of a largest independent set of G. For $X \subseteq V(G)$, we denote by $G[X]$ the subgraph induced by X, and set $G - X := G[V(G) \setminus X]$. For $v \in V(G)$, its open neighborhood is $N_G(v) := \{u \in V(G) : uv \in E(G)\}$. An *induced minor* is obtained by vertex deletions and edge contractions. A graph is *subcubic* if its maximum degree is at most 3. A graph class is *hereditary* if it is closed under taking induced subgraphs.

For a vector x indexed by $V(G)$ with nonnegative entries, we set

$$x(X) := \sum_{v \in X} x_v.$$

A *tree-decomposition* of G consists of a tree T and a family $\{B_t \subseteq V(G) \mid t \in V(T)\}$ that satisfies the following two conditions:

- for every $v \in V(G)$, $T[\{t \in V(T) \mid v \in B_t\}]$ is a nonempty connected subtree of T, and
- for every $uv \in E(G)$, there is some $t \in V(T)$ such that $\{u, v\} \subseteq B_t$.

A set B_t is called a *bag*. The width of the decomposition is $\max_t |B_t| - 1$, and the *treewidth* $\mathrm{tw}(G)$ is the minimum width of a tree-decomposition of G.

In this paper, for convenience, connected components (or components) are vertex sets (not graphs). A *balanced separator* of an n-vertex graph G is a set $S \subseteq V(G)$ such that every connected component of $G - S$ has at most $2n/3$ vertices. By the *components of a separator* S, we mean the connected components of $G - S$. The *separation number* $\mathrm{sn}(G)$ is the least integer s such that every (induced) subgraph of G has a balanced separator of size at most s. Treewidth and separation number are linearly tied. The upper bound on the separation number is due to Robertson and Seymour [47], and the converse is due to Dvořák and Norin [24].

▶ **Theorem 5.** *For every graph G with at least one edge,* $\mathrm{tw}(G) = \Theta(\mathrm{sn}(G))$.

Let $\mathcal{C}$ be a class satisfying the assumption of Theorem 2 for some exponent $0 \leqslant b < 1$. Choose an integer $a \geqslant 2$ such that $b \leqslant 1 - 1/a$. The preceding theorem therefore gives

$$\mathrm{tw}(G) = \mathcal{O}_{\mathcal{C}}(|E(G)|^{1-1/a}) \qquad \text{for every } G \in \mathcal{C}. \tag{2}$$

For edgeless G, the bound is immediate. We fix $\mathcal{C}$ and a for the rest of the proof.

3 Light balanced separators

For a graph G, a nonnegative vector x indexed by $V(G)$, and $0 < \delta < 1$, a balanced separator S of G is (δ, x)-*light* if $x(S) \leqslant \delta x(V(G))$. Note that the balance is measured by the number of vertices in each component, and lightness by the cost x. We set

$$e_G(x) := \sum_{uv \in E(G)} x_u x_v.$$

The goal of this section is to find a (δ, x)-light balanced separator when $x(V(G))$ is sufficiently large and $e_G(x)$ is relatively small.

▶ **Lemma 6.** *Let $\mathcal{C}$ be a graph class with strongly edge-sublinear separators, and let $a \geqslant 2$ be an integer satisfying* (2)*. For every integer $n \geqslant 2$ and rational $0 < \delta < 1$, there is an integer*

$$p = \mathcal{O}_{\mathcal{C}}\big((\delta^{-1} \log n)^{4a-2}\big) \tag{3}$$

such that for any n-vertex graph $G \in \mathcal{C}$ and rational vector $x \in [0,1]^{V(G)}$ with

$$k := x(V(G)) \geqslant p \quad \text{and} \quad e_G(x) \leqslant k^2/p,$$

one can find a (δ, x)-light balanced separator of G. The algorithm is deterministic and takes polynomial time in the input encoding length.

We use a vertex-capacitated version of the Leighton–Rao multicommodity flow–cut theorem [41], with capacities proportional to the costs x_v. Repeated applications yield either the required balanced separator or an all-pairs flow in an induced subgraph. In the latter case, we sample paths from this flow to represent the edges of a subcubic graph of large treewidth. The capacity constraints and the assumptions on x control the sizes of these paths and bound the probability that paths representing nonincident edges intersect or are joined by an edge. With positive probability, discarding long paths leaves a collection whose vertices induce a graph with large treewidth but too few edges to satisfy (2). Heredity ensures that this graph belongs to $\mathcal{C}$, giving a contradiction.

The construction of an *induced almost-embedding* (see Section 3.2 for definitions) from a concurrent flow follows Korhonen and Lokshtanov [39, Section 4.2]. Earlier separator arguments using multicommodity flows include those of Matoušek [42] and Biswal, Lee, and Rao [6]. In particular, the latter already separates long and short flow paths in the treatment of shallow excluded minors: many long paths force large congestion, while restricting to short paths permits a shallow-minor argument. Here we combine the vertex-dependent bounds in (6), the assumptions on x, and a subcubic graph of linear treewidth. These ingredients let us discard edges assigned to long paths while preserving linear treewidth.

3.1 Flows and separations

A *separation* of G is a pair (A, B) of nonempty vertex sets with $A \cup B = V(G)$ and with no edge between $A \setminus B$ and $B \setminus A$. Its separator is $A \cap B$.

We use nonnegative vertex costs x_v, so the cost of a separator S is $x(S) = \sum_{v \in S} x_v$. To find a separator of small cost, we apply a multicommodity flow–cut theorem with vertex capacities proportional to these costs.

Flows and capacity constraints. For $u, v \in V(G)$, let $\mathcal{P}_{u,v}(G)$ be the set of simple paths from u to v, regarded as ordered vertex sequences. In particular, $\mathcal{P}_{v,v}(G) := \{(v)\}$. A *unit all-pairs multicommodity flow* assigns a nonnegative number $f(P)$ to each such path, satisfying

$$\sum_{P \in \mathcal{P}_{u,v}(G)} f(P) = 1 \qquad \text{for every } (u,v) \in V(G)^2.$$

Its *load* at $w \in V(G)$ is

$$\mathrm{load}_f(w) := \sum_{u,v \in V(G)} \sum_{\substack{P \in \mathcal{P}_{u,v}(G) \\ w \in V(P)}} f(P).$$

For a given $\gamma > 0$, the vertex-capacity constraints used below are

$$\mathrm{load}_f(v) \leqslant \gamma x_v \qquad (v \in V(G)).$$

The flow–cut theorem. The following is a consequence of the directed Leighton–Rao flow–cut theorem [41, Theorem 12] and the vertex-splitting reduction [41, Section 3.5]. We adapt the proof to our convention that vertex loads include endpoints and that every ordered pair, including (v, v), has unit demand.

▶ **Lemma 7.** *Given an n-vertex graph G with $n \geqslant 2$, nonnegative rational vertex costs x_v, and a rational $\gamma > 0$, there is a deterministic algorithm, polynomial in the encoding of (G, x, γ), that finds*

1. *a unit all-pairs multicommodity flow f satisfying $\mathrm{load}_f(v) \leqslant \gamma x_v$ for every $v \in V(G)$, or*
2. *a separation (A, B) such that*

$$x(A \cap B) = \mathcal{O}\left(\frac{|A||B|}{\gamma} \log n\right). \tag{4}$$

Proof. Split each vertex v into v^-, v^+ and give the arc $v^- \to v^+$ capacity x_v. Each edge uv gives arcs $u^+ \to v^-$ and $v^+ \to u^-$. These arcs and all reverse arcs $v^+ \to v^-$ receive capacity c, a sufficiently large integer multiple of $\gamma^{-1} n^2 \lceil \log n \rceil$. This finite value acts as infinite capacity: it exceeds both the total demand and the cut-capacity bound below. Run the directed flow–cut algorithm of Leighton and Rao [41, proof of Theorem 12] on this network $\vec{G}$, with demand $1/\gamma$ for every ordered pair of distinct split vertices.

If all demands are routed, retain the u^--to-v^+ flows for all $(u, v) \in V(G)^2$ and decompose them into simple directed paths; the split endpoints are distinct even when $u = v$. Such a path cannot use a reverse arc $w^+ \to w^-$: it reaches w^+ through w^-, so using the reverse arc would repeat a vertex. Every retained path consists of consecutive pairs (w^-, w^+), each traversing the capacity arc $w^- \to w^+$, joined by arcs corresponding to edges of G. Replacing each pair (w^-, w^+) by w gives a simple path in G with the same original endpoints; when $u = v$, this is the singleton path (u). Multiply the flow amount on each path by γ. This yields our unit all-pairs flow with $\mathrm{load}_f(w) \leqslant \gamma x_w$.

Otherwise, we obtain a nontrivial cut $(U, \overline{U})$ of capacity $\mathcal{O}(\gamma^{-1}|U||\overline{U}| \log n) < c$. Put $A := \{v : v^- \in U\}$ and $B := \{v : v^+ \in \overline{U}\}$. No arc of capacity c leaves U, so the reverse arcs and edge arcs ensure that (A, B) is a separation. Its cost $x(A \cap B)$ equals the cut capacity; the bounds $|U| \leqslant 2|A|$ and $|\overline{U}| \leqslant 2|B|$ then give (4). The algorithm in [41, proof of Theorem 12] solves the multicommodity flow linear program and, if the demands cannot be routed, constructs a cut deterministically from an optimal dual solution. The network has polynomial size and rational capacities of polynomial encoding length, so these steps and projecting the flow onto G take polynomial time in the input encoding length. ◀

From separations to a light balanced separator. Given an n-vertex graph G with $n \geqslant 2$ and a nonnegative and nonzero (every entry is nonnegative and not all entries are 0) rational vector x, we will always set $k := x(V(G)) > 0$. For any rational $0 < \delta < 1$, integer ℓ is picked to be at least a sufficiently large universal constant times $\delta^{-1} \log n$, and we set

$$\gamma := \frac{\ell n^2}{3k}. \tag{5}$$

We repeatedly apply Lemma 7 to show the following lemma (as in [10, Lemma 4]).

▶ **Lemma 8.** *With $G, x, \delta, \ell, \gamma$ chosen as above, there is a deterministic algorithm, polynomial in their encoding length, that finds*

1. *a (δ, x)-light balanced separator of G, or*
2. *an induced subgraph G' with more than $2n/3$ vertices and a probability distribution on the ordered simple paths of G'. If Q is drawn from this distribution, then*

$$\Pr[Q \in \mathcal{P}_{u,v}(G')] = \frac{1}{|V(G')|^2} \qquad \text{for every } (u, v) \in V(G')^2.$$

Moreover,

$$\Pr[v \in V(Q)] \leqslant \frac{\ell x_v}{k} \qquad \text{for every } v \in V(G'). \tag{6}$$

Proof. Start with $G' := G$ and $S := \varnothing$. While $|V(G')| > 2n/3$, apply Lemma 7 to G', using the restriction of x and the fixed parameter γ. Stop if a flow is returned. Otherwise, swap the sides of the returned separation (A_i, B_i) if necessary so that $|A_i| \geqslant |B_i|$. Add $A_i \cap B_i$ to S and replace G' by $G'[A_i \setminus B_i]$. Each such step removes the nonempty set B_i, so these sets are pairwise disjoint and there are at most n calls to Lemma 7.

By (4), the total separator cost is

$$x(S) = \mathcal{O}\left(\frac{\log n}{\gamma} \sum_i |A_i||B_i|\right) = \mathcal{O}\left(\frac{n^2 \log n}{\gamma}\right) = \mathcal{O}\left(\frac{3k \log n}{\ell}\right) \leqslant \delta k. \tag{7}$$

Here $|A_i| \leqslant n$ and $\sum_i |B_i| \leqslant n$ give the second bound, and the choice of ℓ gives the last inequality. If every call returns a separation, the discarded sets $B_i \setminus A_i$ and the final set $V(G')$ partition $V(G) \setminus S$, with no edges between distinct parts. Since $|A_i \setminus B_i| \geqslant |B_i \setminus A_i|$, each discarded set has at most $n/2$ vertices, and the final set has at most $2n/3$ vertices. Thus S is a (δ, x)-light balanced separator.

If a unit all-pairs flow f is returned on G', define Q by the following sampling rule. Choose u, v independently and uniformly from $V(G')$. Conditional on this choice, select a path $P \in \mathcal{P}_{u,v}(G')$ with probability $f(P)$, and set $Q := P$. The unit-flow equations ensure that these conditional probabilities sum to 1. Consequently, every ordered path P in G' satisfies

$$\Pr[Q = P] = \frac{f(P)}{|V(G')|^2}.$$

Thus, for every vertex $w \in V(G')$,

$$\Pr[w \in V(Q)] = \frac{\mathrm{load}_f(w)}{|V(G')|^2} \leqslant \frac{\gamma x_w}{|V(G')|^2} \leqslant \frac{\ell x_w}{k},$$

where the last inequality uses $|V(G')| > 2n/3$ and $(3/2)^2 < 3$. This gives item 2. Its distribution is specified deterministically by the rational probabilities $f(P)/|V(G')|^2$ on the polynomially many paths carrying a positive flow. The running time follows from Lemma 7 and the bound on the number of iterations. ◀

3.2 Treewidth and sampled paths

We rule out item 2 of Lemma 8 by sampling paths for the edges of a t-vertex subcubic graph of treewidth $\Omega(t)$, following Korhonen and Lokshtanov [39, Section 4.2]. Using (6), we bound the probability that paths assigned to nonincident edges intersect or are joined by an edge. Summing (6) over vertices also bounds the expected number of vertices on each sampled path. After deleting the edges assigned to long paths, the remaining path vertices induce a graph F with treewidth $\Omega(t)$ and $\mathcal{O}(t\ell^2)$ edges. For a suitable choice of t, these bounds contradict (2).

An *almost-embedding* of H in G is a pair (ϕ, π), where $\phi : V(H) \to V(G)$ and π assigns to each edge uv of H a path from $\phi(u)$ to $\phi(v)$ in G, such that paths assigned to *nonincident* edges (i.e., with no common endpoint) are vertex-disjoint. The almost-embedding is *induced* if there is also no edge of G between paths assigned to nonincident edges. The map ϕ need not be injective, and singleton paths are allowed images of π.

The following observation transfers a tree-decomposition through an almost-embedding.

▶ **Lemma 9.** *Let H have maximum degree at most d and no isolated vertices. If H has an almost-embedding in G, then every tree-decomposition of G with bags of size at most w gives a tree-decomposition of H with bags of size at most $2dw$.*

Proof. Let (ϕ, π) be an almost-embedding of H in G. For each $v \in V(G)$, let L_v consist of the endpoints of the edges $e \in E(H)$ such that $v \in V(\pi(e))$. We first claim that $|L_v| \leqslant 2d$. Indeed, if some path $\pi(xy)$ contains v, every edge whose path contains v shares an endpoint with xy, since paths assigned to nonincident edges are vertex-disjoint. All endpoints of these edges belong to $N_H(x) \cup N_H(y)$. This union has size at most $2d$.

Take a tree-decomposition $(T, (B_t)_{t \in V(T)})$ of G and replace each bag B_t by

$$B'_t := \bigcup_{v \in B_t} L_v.$$

Every vertex of H belongs to a new bag, since it has an incident edge. Both endpoints of an edge $xy \in E(H)$ belong to every new bag whose original bag intersects $\pi(xy)$. For a vertex x of H, the union of the paths representing edges incident with x is connected, since all of them contain $\phi(x)$. In a tree-decomposition, the bags meeting a connected vertex set form a connected subtree: the subtrees for adjacent vertices intersect in a bag containing their edge. Thus the bags B'_t containing x form a connected subtree. They give a tree-decomposition of H, and $|B'_t| \leqslant 2d|B_t|$ proves the bound. ◀

To apply this observation, we use the following existence result.

▶ **Lemma 10** ([37], [31, Proposition 1])**.** *For every sufficiently large integer t, there is a t-vertex subcubic graph of treewidth $\Omega(t)$.*

Subcubic expanders exist [37], and their expansion implies linear treewidth [31, Proposition 1]. We can now show the main novel technical lemma of this section.

▶ **Lemma 11.** *For every $\ell \geqslant 1$, there is an integer $p = \mathcal{O}_{\mathcal{C}}(\ell^{4a-2})$ with the following property. Let $G \in \mathcal{C}$ and $x \in [0,1]^{V(G)}$, and put $k := x(V(G))$. If $k \geqslant p$ and $e_G(x) \leqslant k^2/p$, then G does not satisfy item 2 of Lemma 8.*

Proof. Suppose that item 2 of Lemma 8 holds, with induced subgraph G' and random path Q. Choose an integer t that is a sufficiently large constant multiple of ℓ^{2a-2}, and let H be the graph yielded by Lemma 10. We will specify how large this multiple needs to be at

the end of the proof. We then choose an integer $p = \mathcal{O}(t^2\ell^2)$ with a sufficiently large implicit constant, which gives the required bound on p.

Fix an orientation of each edge of H. Choose $\phi(u)$, for $u \in V(H)$, independently and uniformly from $V(G')$. Conditional on ϕ, choose the paths $\pi(uv)$ independently over the oriented edges uv, with

$$\Pr[\pi(uv) = P \mid \phi] = |V(G')|^2 \Pr[Q = P] \qquad \text{for } P \in \mathcal{P}_{\phi(u),\phi(v)}(G').$$

These probabilities sum to 1 because the ordered endpoints of Q are independent and uniform in $V(G')$. Each $\pi(uv)$ therefore has the same distribution as Q. For nonincident edges, the endpoint assignments are independent, and the paths are sampled independently conditional on those assignments. Hence the two paths are independent.

Let e, f be nonincident edges of H. By the union bound and independence,

$$\begin{aligned}\Pr[V(\pi(e)) \cap V(\pi(f)) \neq \varnothing] &\leqslant \sum_{v \in V(G')} \Pr[v \in V(\pi(e))] \cdot \Pr[v \in V(\pi(f))] \\ &\leqslant \frac{\ell^2}{k^2} \sum_{v \in V(G')} x_v^2 \\ &\leqslant \frac{\ell^2}{k^2} \sum_{v \in V(G)} x_v = \frac{\ell^2}{k}, \end{aligned} \tag{8}$$

where we use (6), $x_v^2 \leqslant x_v$, and $x(V(G)) = k$. Moreover,

$$\begin{aligned}&\Pr[\text{an edge joins } V(\pi(e)) \text{ and } V(\pi(f))] \\ &\quad\leqslant \sum_{uv \in E(G')} \Big(\Pr[u \in V(\pi(e))] \cdot \Pr[v \in V(\pi(f))] + \Pr[v \in V(\pi(e))] \cdot \Pr[u \in V(\pi(f))]\Big) \\ &\quad\leqslant \frac{2\ell^2}{k^2} \sum_{uv \in E(G')} x_u x_v \\ &\quad\leqslant \frac{2\ell^2}{k^2} \sum_{uv \in E(G)} x_u x_v \leqslant \frac{2\ell^2}{p}. \end{aligned} \tag{9}$$

The two summands account for the two choices of which path contains which endpoint of an edge. We again use the union bound, independence, and (6); the last inequality follows from the lemma assumption that $e_G(x) \leqslant k^2/p$. Since $|E(H)| \leqslant 3t/2$ and $k \geqslant p$, the probability of any intersection or adjacency between paths representing nonincident edges is

$$\mathcal{O}\left(t^2\left(\frac{\ell^2}{k} + \frac{2\ell^2}{p}\right)\right) = \mathcal{O}(t^2\ell^2/p) \leqslant 1/4, \tag{10}$$

by our choice of p.

Summing (6) over vertices gives

$$\mathbb{E}[|V(\pi(e))|] \leqslant \frac{\ell}{k} \sum_{v \in V(G')} x_v \leqslant \ell.$$

By linearity of expectation and Markov's inequality,

$$\Pr\left[\sum_{e \in E(H)} |V(\pi(e))| \leqslant 4\ell|E(H)|\right] \geqslant \frac{3}{4}.$$

Fix a universal constant c (not depending on t) such that $8|E(H)| \leqslant c \cdot \mathrm{tw}(H)$.

Whenever $\sum_{e \in E(H)} |V(\pi(e))| \leqslant 4\ell|E(H)|$, dividing the right-hand side by $c\ell$ shows that at most $4|E(H)|/c \leqslant \mathrm{tw}(H)/2$ paths have more than $c\ell$ vertices. Together with (10), this gives positive probability that (ϕ, π) is an induced almost-embedding of H in G' and that at most $\mathrm{tw}(H)/2$ paths exceed this size (indeed, both events have probability at least 3/4). Fix such an outcome.

Delete every edge e of H for which $|V(\pi(e))| > c\ell$, and remove the resulting isolated vertices, obtaining H'. Adding one edge to a graph increases its treewidth by at most 1: add one endpoint of that edge to every bag of a tree-decomposition. Removing isolated vertices preserves positive treewidth. Consequently,

$$\mathrm{tw}(H') \geqslant \mathrm{tw}(H)/2 = \Omega(t). \tag{11}$$

Let

$$F := G\left[\bigcup_{e \in E(H')} V(\pi(e))\right].$$

Restricting ϕ and π to H' gives an induced almost-embedding of H' in F.

Every retained path has at most $c\ell$ vertices. Since the almost-embedding is induced, every edge of F has both endpoints on one retained path or joins paths assigned to incident edges of H'. There are at most $3t/2$ retained paths and at most $3t$ pairs of incident edges, since H' is subcubic. Each path or pair of paths contains $\mathcal{O}(\ell)$ vertices and therefore accounts for at most $\mathcal{O}(\ell^2)$ edges in F. Hence

$$|E(F)| = \mathcal{O}(t\ell^2). \tag{12}$$

On the other hand, Lemma 9 with $d = 3$ and (11) give $\mathrm{tw}(F) = \Omega(t)$. The graph F is induced in G and hence belongs to $\mathcal{C}$. Applying (2), we obtain

$$t = \mathcal{O}_{\mathcal{C}}\big((t\ell^2)^{1-1/a}\big), \qquad \text{and thus} \qquad t = \mathcal{O}_{\mathcal{C}}(\ell^{2a-2}).$$

Choosing the constant in the definition of t larger than the last implicit constant gives a contradiction. The choices of t and then p depend only on $\mathcal{C}$ and ℓ, as required. ◀

We can now conclude.

Proof of Lemma 6. Choose $\ell := \lceil c_0 \delta^{-1} \log n \rceil$, where c_0 is a sufficiently large universal constant for Lemma 8, and set γ by (5). Let p be the integer given by Lemma 11. Then $p = \mathcal{O}_{\mathcal{C}}(\ell^{4a-2})$, which gives (3). Apply Lemma 8. Item 2 contradicts Lemma 11, so item 1 gives a (δ, x)-light balanced separator. The sampling in Lemma 11 existentially shows that item 2 cannot hold, but will not be used in the overall algorithm; from this section, we will only rely on the deterministic algorithm of Lemma 8. ◀

Increasing p strengthens the assumptions of Lemma 6, so the conclusion holds for every larger threshold. By (3), we may therefore choose p nondecreasing in n.

4 A linear relaxation and conditioning

To apply Lemma 6 to a target $k \geqslant p$, we need vertex costs $x_v \in [0, 1]$ satisfying

$$x(V(G)) = k \qquad \text{and} \qquad e_G(x) \leqslant k^2/p.$$

We obtain these costs from a feasible point of a linear relaxation by conditioning on the inclusion of a small vertex set.

Fix an n-vertex graph G, an integer $1 \leqslant k \leqslant n$, and an integer $r \geqslant 1$. Introduce a variable y_A for every $A \subseteq V(G)$ with $|A| \leqslant r+2$. Consider the following system of linear constraints.

$$\begin{aligned} y_\varnothing &= 1, \\ 0 \leqslant y_A &\leqslant 1 \qquad (|A| \leqslant r+2), \\ y_{A\cup\{v\}} &\leqslant y_A \qquad (|A| \leqslant r+1,\ v \in V(G)), \\ y_A &= 0 \qquad (|A| \leqslant r+2,\ A \text{ is not an independent set}), \\ \sum_{v\in V(G)} y_{A\cup\{v\}} &= k y_A \quad (|A| \leqslant r+1). \end{aligned} \tag{13}$$

Let $R_r(G,k)$ be the polytope defined by (13). Every independent set I of size k gives a point of this polytope by setting

$$y_A := \mathbf{1}[A \subseteq I].$$

The first four rows hold directly. For the last row, observe that

$$\sum_{v\in V(G)} \mathbf{1}[A\cup\{v\} \subseteq I] = \mathbf{1}[A \subseteq I] \sum_{v\in V(G)} \mathbf{1}[v \in I] = k \cdot \mathbf{1}[A \subseteq I].$$

The indexed variables replace products of 0–1 indicators, as in the Sherali–Adams construction [48].

The number of variables is

$$d := \sum_{j=0}^{\min\{n,r+2\}} \binom{n}{j} = n^{\mathcal{O}(r)}. \tag{14}$$

There are $n^{\mathcal{O}(r)}$ constraints, which can all be checked in that time.

4.1 Conditioning

Fix a rational vector $y \in R_r(G,k)$. For $A \subseteq V(G)$ with $|A| \leqslant r$ and $y_A > 0$, define

$$x_v := \frac{y_{A\cup\{v\}}}{y_A}.$$

The constraints $0 \leqslant y_{A\cup\{v\}} \leqslant y_A$ in (13) give $0 \leqslant x_v \leqslant 1$, and the last row of (13) gives

$$\sum_{v\in V(G)} x_v = \frac{1}{y_A} \sum_{v\in V(G)} y_{A\cup\{v\}} = k.$$

Thus the numbers x_v/k define a probability distribution on $V(G)$. For a vertex v with $x_v > 0$, adding v to A preserves the positivity of the denominator, since $y_{A\cup\{v\}} > 0$. We call adding v to A and recomputing the ratios *conditioning* on v. Conditioning on a small set of variables is used in the rounding algorithms of Barak, Raghavendra, and Steurer [5] and Raghavendra and Tan [46].

▶ **Lemma 12.** *Let $p \geqslant 1$ be an integer and let $y \in R_r(G,k)$ be rational. If $r \geqslant 2p\log n$, then a deterministic algorithm finds a set $A \subseteq V(G)$, with $|A| \leqslant r$ and $y_A > 0$, such that the vector*

$$x := \left(\frac{y_{A\cup\{v\}}}{y_A}\right)_{v\in V(G)}$$

satisfies

$$x \in [0,1]^{V(G)}, \qquad x(V(G)) = k, \qquad e_G(x) \leqslant k^2/p.$$

Its running time is polynomial in r and in the encoding lengths of y and p.

Proof. We first prove that a suitable set A exists. Fix A with $|A| < r$ and $y_A > 0$, and write

$$\pi_u := \frac{x_u}{k}, \qquad q_{uv} := \frac{y_{A\cup\{u,v\}}}{k^2 y_A}.$$

Applying the last row of (13) at $A\cup\{v\}$ gives

$$\sum_u q_{uv} = \frac{1}{k^2 y_A}\sum_u y_{A\cup\{u,v\}} = \frac{y_{A\cup\{v\}}}{k y_A} = \pi_v. \tag{15}$$

By symmetry, $\sum_v q_{uv} = \pi_u$. Therefore q is a probability distribution on $V(G)^2$, with both marginals equal to π. For each v with $\pi_v > 0$, let π^v be the distribution associated with $A\cup\{v\}$. Its coordinates are

$$\pi^v_u := \frac{y_{A\cup\{u,v\}}}{k y_{A\cup\{v\}}} = \frac{q_{uv}}{\pi_v}. \tag{16}$$

Our proof uses an entropy-decrease argument similar to that of Raghavendra and Tan [46, Section 4], applied here to the distribution π on $V(G)$. We show that, whenever $e_G(x) > k^2/p$, conditioning on some vertex decreases entropy by more than $2/p$. Since entropy lies between 0 and $\log n$, this bounds the number of successive choices needed to obtain a suitable set A. For a probability distribution π on $V(G)$, its *entropy* is

$$H(\pi) := -\sum_{u:\pi_u>0} \pi_u \log \pi_u.$$

Every summand is nonnegative, since $0 < \pi_u \leqslant 1$, so $H(\pi) \geqslant 0$. For the upper bound, Jensen's inequality for the concave function log, with weights π_u summing to 1, gives

$$\begin{aligned} H(\pi) &= \sum_{u:\pi_u>0} \pi_u \log \frac{1}{\pi_u} \\ &\leqslant \log\left(\sum_{u:\pi_u>0} \pi_u \frac{1}{\pi_u}\right) = \log\bigl|\{u : \pi_u > 0\}\bigr| \leqslant \log n. \end{aligned}$$

Expanding the entropy definitions, we obtain

$$\begin{aligned} H(\pi) - \sum_{v:\pi_v>0} \pi_v H(\pi^v) &= -\sum_{u:\pi_u>0} \pi_u \log \pi_u + \sum_{u,v:q_{uv}>0} q_{uv} \log \frac{q_{uv}}{\pi_v} \\ &= \sum_{u,v:q_{uv}>0} q_{uv}\left(\log \frac{q_{uv}}{\pi_v} - \log \pi_u\right) \\ &= \sum_{u,v:q_{uv}>0} q_{uv} \log \frac{q_{uv}}{\pi_u \pi_v}. \end{aligned} \tag{17}$$

The first equality uses (16), and the second uses the marginal identity $\sum_v q_{uv} = \pi_u$ from (15) and symmetry. The sum on the left averages the entropies after conditioning on a vertex, with weights π_v.

Since the positive entries of q sum to 1, Jensen's inequality gives

$$\sum_{u,v:q_{uv}>0} q_{uv} \log \frac{q_{uv}}{\pi_u \pi_v} = - \sum_{u,v:q_{uv}>0} q_{uv} \log \frac{\pi_u \pi_v}{q_{uv}} \geqslant - \log \left(\sum_{u,v:q_{uv}>0} \pi_u \pi_v \right).$$

If $uv \in E(G)$, then $q_{uv} = 0$ by (13). Thus, since $\sum_{u,v} \pi_u \pi_v = (\sum_u \pi_u)(\sum_v \pi_v) = 1 \cdot 1 = 1$,

$$\begin{aligned} \sum_{u,v:q_{uv}>0} \pi_u \pi_v &= 1 - \sum_{u,v:q_{uv}=0} \pi_u \pi_v \\ &\leqslant 1 - \sum_{uv \in E(G)} (\pi_u \pi_v + \pi_v \pi_u) \\ &= 1 - \frac{2}{k^2} \sum_{uv \in E(G)} x_u x_v = 1 - \frac{2e_G(x)}{k^2}. \end{aligned}$$

The two summands for each edge account for the ordered pairs (u, v) and (v, u). Combining these inequalities yields

$$H(\pi) - \sum_{v:\pi_v>0} \pi_v H(\pi^v) \geqslant - \log \left(1 - \frac{2e_G(x)}{k^2} \right) \geqslant \frac{2e_G(x)}{k^2}. \tag{18}$$

The last inequality uses $-\log(1-t) \geqslant t$ for $0 \leqslant t < 1$. The logarithm in (18) has a positive argument, since $1 - 2e_G(x)/k^2 \geqslant \sum_{u,v:q_{uv}>0} \pi_u \pi_v > 0$. The sum is positive because q has total mass 1 and $q_{uv} > 0$ implies $\pi_u \pi_v > 0$. Since the weights π_v sum to 1, (18) implies that if $e_G(x) > k^2/p$, some vertex v with $\pi_v > 0$ satisfies

$$H(\pi^v) < H(\pi) - 2/p.$$

Start with $A := \varnothing$ and, while the required inequality fails, add such a vertex v. Every denominator remains positive, and each choice increases $|A|$ by 1. Since entropy starts at most $\log n$ and remains nonnegative, fewer than r such choices are possible under the assumption $r \geqslant 2p \log n$. Hence a set A satisfying the conclusion exists.

To find one deterministically, examine all sets $A \subseteq V(G)$ of size at most r with $y_A > 0$, and test

$$p \sum_{uv \in E(G)} y_{A \cup \{u\}} y_{A \cup \{v\}} \leqslant k^2 y_A^2.$$

This is equivalent to $e_G(x) \leqslant k^2/p$. Upon finding a set A that passes the test, set $x_v := y_{A \cup \{v\}}/y_A$ for every $v \in V(G)$ and return (A, x). All required coordinates occur in y, and there are at most d candidate sets, where d is given by (14). Each test uses $\mathcal{O}(n^2)$ rational operations on numbers of polynomial encoding length. This proves the running-time bound. ◀

5 The approximation algorithm

We combine Lemmas 6 and 12 in a recursive algorithm. A call on G returns an independent set I_G and an integer $u_G \geqslant \alpha(G)$, with $|I_G|$ approximating u_G. For each k (guess of the size of a maximum independent set), the outputs (I_C, u_C) from the components of a balanced separator either yield a set of size close to k or a valid inequality violated by the current query, which we can transmit to the ellipsoid algorithm. We first establish these inequalities and explain their use in the overall algorithm.

5.1 Feasibility and valid inequalities

For a graph G, an integer $r \geqslant 1$, and an integer target k, let d be as in (14). We write

$$z = (z_A)_{A \subseteq V(G),\, |A| \leqslant r+2} \in \mathbb{R}^d$$

for a variable vector and $y \in \mathbb{Q}^d$ for the query point. Define

$$\mathcal{P}_r(G,k) := \operatorname{conv}\left\{ \left(\mathbf{1}[A \subseteq I]\right)_{\substack{A \subseteq V(G) \\ |A| \leqslant r+2}} : I \text{ is an independent set of } G,\ |I| = k \right\}. \tag{19}$$

Here conv denotes the convex hull, with $\operatorname{conv} \varnothing = \varnothing$. Thus

$$\mathcal{P}_r(G,k) \neq \varnothing \iff \alpha(G) \geqslant k, \qquad \mathcal{P}_r(G,k) \subseteq R_r(G,k). \tag{20}$$

An inequality is *valid for* $\mathcal{P}_r(G,k)$ if every point of this polytope satisfies it. We already observed that the constraints in (13) are valid for $\mathcal{P}_r(G,k)$. The following lemma gives further such inequalities from upper bounds on the independence numbers of induced subgraphs.

▶ **Lemma 13.** *Let $C \subseteq V(G)$ and $u_C \geqslant \alpha(G[C])$. For every $A \subseteq V(G)$ with $|A| \leqslant r$, every vector $z \in \mathcal{P}_r(G,k)$ satisfies*

$$\sum_{v \in C} z_{A \cup \{v\}} \leqslant u_C z_A. \tag{21}$$

A point $y \in \mathbb{R}^d$ with $y_A > 0$ violates this inequality whenever $\sum_{v \in C} y_{A \cup \{v\}} / y_A > u_C$.

Proof. Let I be an independent set of G of size k, and evaluate (21) at the corresponding vector in (19). If $A \not\subseteq I$, both sides of (21) are 0. If $A \subseteq I$, the inequality reads $|I \cap C| \leqslant u_C$, which holds since $I \cap C$ is an independent set of $G[C]$. By linearity, the inequality holds for every convex combination of these vectors, hence throughout $\mathcal{P}_r(G,k)$. For $y_A > 0$, dividing (21) by y_A shows that it is violated precisely when

$$\sum_{v \in C} \frac{y_{A \cup \{v\}}}{y_A} > u_C. \qquad \blacktriangleleft$$

A *strong separation oracle* for a rational polytope $Q \subseteq \mathbb{R}^d$, on input $y \in \mathbb{Q}^d$, either certifies $y \in Q$ or returns a linear inequality valid for Q and violated by y. We use the exact rational ellipsoid feasibility algorithm of Grötschel, Lovász, and Schrijver [32, Theorem 6.4.1]. Given such an oracle and a bound on the encoding length of each inequality in a description of Q, it finds a point of Q or certifies emptiness. This guarantee also applies when the polytope is contained in a proper affine subspace. The number of oracle calls and the additional work are polynomial in d and the encoding bound. Here $\mathcal{P}_r(G,k) \subseteq [0,1]^d$ has 0–1 vertices of encoding length $\mathcal{O}(d)$, so a polynomial bound in d on the encoding length of each defining inequality follows from [32, Lemma 6.2.4].

We run the ellipsoid algorithm for $\mathcal{P}_r(G,k)$, with an early stopping rule. Each query y is passed to RoundOrCut(G,k,y). This procedure returns either a sufficiently large (approximate) independent set, in which case we stop, or an inequality valid for $\mathcal{P}_r(G,k)$ and violated by y, in which case the ellipsoid algorithm continues.

Section 5.2 proves that the procedure always returns one of these two answers. Consequently, until an independent set is returned, every answer is a separation for the same fixed polytope. The exact feasibility guarantee then gives either an independent set or a certificate that $\mathcal{P}_r(G,k)$ is empty. This is the *round-or-cut* approach introduced by Carr et al. [11]. Chakrabarty and Negahbani [12, Section 3.2] similarly use an auxiliary optimization problem to obtain either an approximate solution or an inequality separating the current query from a fixed polytope.

5.2 Algorithm and correctness

On an edgeless input, return all vertices. Otherwise, let $n \geqslant 2$ be the number of vertices of the original input, and fix rational $0 < \varepsilon < 1$. Put

$$\lambda := \lceil \log_{3/2} n \rceil, \qquad \delta := \frac{\varepsilon}{\lambda}. \tag{22}$$

Choose the integer threshold p given by Lemma 6 for n and δ, and set $r := 2p\lambda$. We use the nondecreasing choice of p specified after Lemma 11. Thus the same threshold applies to every recursive induced subgraph, and $r \geqslant 2p \log n$ satisfies Lemma 12. All these parameters remain fixed throughout the recursion, with $p = \mathcal{O}_{\mathcal{C}}\big((\lambda/\delta)^{4a-2}\big)$.

The procedure Solve. For an integer $t \geqslant 1$, put $h(t) := \lceil \log_{3/2} t \rceil$, and set $h(0) := 0$. We define a procedure Solve(G) that returns an independent set I_G and an integer u_G satisfying

$$\alpha(G) \leqslant u_G \leqslant |V(G)|, \qquad |I_G| \geqslant (1-\delta)^{h(|V(G)|)} u_G. \tag{23}$$

Each recursive call from G is on $G[C]$ with $|C| \leqslant 2|V(G)|/3$, where C is a component left after deleting a balanced separator. Thus $h(|V(G)|)$ bounds the recursion depth.

The procedure considers the targets $k = |V(G)|, |V(G)| - 1, \ldots, 1$. For each target, it either returns an independent set of size at least

$$(1-\delta)^{h(|V(G)|)} k,$$

or proves $\alpha(G) < k$. For $k \leqslant p$, it searches exhaustively for an independent set of size k; for $k > p$, it runs the ellipsoid algorithm with queries handled by RoundOrCut, described below. At the first successful target k, the preceding rejections give $\alpha(G) \leqslant k$, so the procedure returns $u_G := k$. A returned independent set establishes the approximation bound for the current target. The pseudocode is given in Algorithm 1.

The procedure RoundOrCut. For a target $k > p$, the procedure RoundOrCut(G, k, y) first checks (13) and returns a violated inequality if one exists. When a query violates an equality in (13), it returns the violated one of its two associated inequalities. For a query satisfying (13), it obtains (A, x) by Lemma 12 and finds a (δ, x)-light balanced separator S by Lemma 6. It then calls Solve$(G[C])$ to obtain (I_C, u_C) for each component C of $G - S$.

Each recursive call on $G[C]$ uses the shared parameters. The parent computes $x(C) = \sum_{v \in C} y_{A \cup \{v\}}/y_A$ for its query y and compares it with the returned bound u_C. If $x(C) > u_C$ for some component, the parent returns (21) to the ellipsoid algorithm for $\mathcal{P}_r(G, k)$; otherwise, it returns the union of the sets I_C. The pseudocode is given in Algorithm 2.

Algorithm 1 SOLVE(G)

Input: An induced subgraph G of the input.
Data: The shared parameters λ, δ, p, r, fixed for the original input.
Output: A pair (I_G, u_G) satisfying (23), with I_G an independent set of G.
if $E(G) = \varnothing$ **then**
  return $(V(G), |V(G)|)$
for $k = |V(G)|, |V(G)| - 1, \ldots, 1$ **do**
  if $k \leqslant p$ **then**
    foreach $I \subseteq V(G)$ *with* $|I| = k$ **do**
      if I *is independent* **then**
        return (I, k)
  else
    Run the exact ellipsoid feasibility algorithm for $\mathcal{P}_r(G, k)$
    Handle each oracle query or feasible output y as follows:
      Call ROUNDORCUT(G, k, y)
      if *it returns an independent set* I **then**
        Stop the ellipsoid algorithm and **return** (I, k)
      else
        Pass the returned inequality to the ellipsoid algorithm and resume it
    Once the ellipsoid algorithm certifies emptiness, continue with the next target

Algorithm 2 ROUNDORCUT(G, k, y)

Input: An induced subgraph G of the input, an integer $k > p$, and a rational query vector y, with one coordinate y_A for each $A \subseteq V(G)$ with $|A| \leqslant r + 2$.
Data: The shared parameters λ, δ, p, r, fixed for the original input.
Output: An independent set I with $|I| \geqslant (1 - \delta)^{h(|V(G)|)} k$, or an inequality valid for the polytope $\mathcal{P}_r(G, k)$ defined by (19) and violated by y.
if y *violates* (13) **then**
  return a violated inequality from (13)
Obtain (A, x) by Lemma 12 with parameter p
Obtain a (δ, x)-light balanced separator S by Lemma 6
$I \leftarrow \varnothing$
foreach *component* C *of* $G - S$ **do**
  $(I_C, u_C) \leftarrow$ SOLVE$(G[C])$
  if $x(C) > u_C$ **then**
    return $\sum_{v \in C} z_{A \cup \{v\}} \leqslant u_C z_A$
  $I \leftarrow I \cup I_C$
return I

▶ **Lemma 14.** *Every call* Solve(G) *terminates and satisfies* (23).

Proof. We argue by induction on $n' := |V(G)|$. The edgeless case, including the empty graph, is immediate. Otherwise, every recursive call is on an induced subgraph with at most $2n'/3 < n'$ vertices, so the induction hypothesis applies to it.

Consider $k > p$ and a query y generated by the ellipsoid algorithm for $\mathcal{P}_r(G,k)$. If y violates (13), the returned inequality is valid for $\mathcal{P}_r(G,k)$ by (20). Otherwise, conditioning returns A with $y_A > 0$ and a vector $x \in [0,1]^{V(G)}$ satisfying $x(V(G)) = k$ and $e_G(x) \leqslant k^2/p$. Since $k > p$, Lemma 6 applies and returns a balanced separator S of G with $x(S) \leqslant \delta k$. For every component C of $G - S$, induction gives $u_C \geqslant \alpha(G[C])$. If $x(C) > u_C$, Lemma 13 therefore gives a valid violated inequality for the fixed polytope $\mathcal{P}_r(G,k)$.

Suppose that every component satisfies $x(C) \leqslant u_C$. The union I of the sets I_C is independent since there are no edges between distinct components of $G - S$. Since S is balanced, every nonempty component satisfies $|C| \leqslant 2n'/3$, and hence $h(|C|) \leqslant h(n') - 1$. Thus the induction hypothesis yields

$$\begin{aligned} |I| = \sum_C |I_C| &\geqslant (1-\delta)^{h(n')-1} \sum_C u_C \\ &\geqslant (1-\delta)^{h(n')-1} \sum_C x(C) \\ &= (1-\delta)^{h(n')-1}(k - x(S)) \\ &\geqslant (1-\delta)^{h(n')} k. \end{aligned} \tag{24}$$

The last inequality uses the fact that S is (δ, x)-light. Thus RoundOrCut terminates and returns either an independent set of size at least $(1-\delta)^{h(n')}k$ or a valid violated inequality. By the remarks in Section 5.1, for each $k > p$, the ellipsoid algorithm with the early stopping rule terminates with an independent set or a certificate that $\mathcal{P}_r(G,k) = \varnothing$.

For $k \leqslant p$, we only reject the target k if exhaustive search finds no independent set of size k, so $\alpha(G) < k$. For $k > p$, rejection means that the ellipsoid algorithm certifies $\mathcal{P}_r(G,k) = \varnothing$, which also implies $\alpha(G) < k$ by (20). Every rejected target therefore is larger than $\alpha(G)$. Since the computation for each target terminates, the loop must return an independent set by the time it reaches $k = \alpha(G)$.

Let k be the first successful run. If $k = |V(G)|$, then $\alpha(G) \leqslant k$ directly. Otherwise, target $k+1$ was rejected, so $\alpha(G) < k+1$, and integrality gives $\alpha(G) \leqslant k$. Thus returning $u_G := k$ gives the required upper bound. The size guarantee follows from (24) when $k > p$, and from $|I_G| = k$ otherwise. ◀

▶ **Corollary 15.** *Let $G \in \mathcal{C}$ and let $0 < \varepsilon < 1$ be rational. With the parameter choices above,* Solve(G) *returns an independent set I_G satisfying $|I_G| \geqslant (1-\varepsilon)\alpha(G)$.*

Proof. The edgeless case is immediate. Otherwise, $h(n) = \lambda$, so Lemma 14 and Bernoulli's inequality give

$$|I_G| \geqslant (1-\delta)^{h(n)} u_G \geqslant (1 - \delta h(n))\alpha(G) = (1-\varepsilon)\alpha(G).$$ ◀

5.3 Running time and rational arithmetic

For the time analysis, we may assume that ε has encoding length $\mathcal{O}(\log \varepsilon^{-1})$. Indeed, a rational ε can be rounded down to a dyadic number ε' with $\varepsilon/2 < \varepsilon' \leqslant \varepsilon$, in polynomial time in its encoding length. Using ε' preserves the requested guarantee and changes the final bound by at most a constant factor in the exponent.

All constants below may depend on $\mathcal{C}$. The parameter choices in Section 5.2 give

$$p = \mathcal{O}_{\mathcal{C}}(\varepsilon^{-(4a-2)}\lambda^{8a-4}), \qquad r = \mathcal{O}(p\lambda). \tag{25}$$

In every recursive call, the dimension in (14) and the number of constraints in (13) are at most $n^{\mathcal{O}(r)}$, where n is the original input size. Section 5.1 establishes a polynomial bound on the encoding length of each inequality in some description of $\mathcal{P}_r(G,k)$. Every returned inequality has integer coefficients of absolute value $\mathcal{O}(n)$: this holds directly for (13), and for (21) it follows from $u_C \leqslant |C|$, after repeated coordinates are collected. Consequently, over all targets in one call to Solve, the ellipsoid algorithm makes at most $n^{\mathcal{O}(r)}$ queries and uses $n^{\mathcal{O}(r)}$ additional time. Its rational queries have encoding length $n^{\mathcal{O}(r)}$.

Processing one query takes $n^{\mathcal{O}(r)}$ time, excluding the time spent in recursive calls to Solve. This includes checking (13), enumerating the conditioning sets in Lemma 12, and applying the separator algorithm of Lemma 6. The coordinates of x are ratios of entries of the current query, so their encoding lengths are polynomial in the encoding length of that query. All comparisons are performed by exact rational arithmetic.

Enumeration for targets $k \leqslant p$ takes $n^{\mathcal{O}(p)}$ time, which is absorbed in $n^{\mathcal{O}(r)}$. A call to Solve therefore performs $n^{\mathcal{O}(r)}$ work outside its recursive calls and makes at most $n^{\mathcal{O}(r)}$ recursive calls: there are $n^{\mathcal{O}(r)}$ queries, each generating at most n child calls. Each child has at most two thirds as many vertices as its parent, so the recursion depth is at most λ. The total running time is consequently

$$n^{\mathcal{O}(r\lambda)} = 2^{\mathcal{O}(r\lambda^2)} = 2^{\mathcal{O}_{\mathcal{C}}(p\lambda^3)} = 2^{\mathcal{O}_{\mathcal{C}}(\varepsilon^{-(4a-2)}(\log n)^{8a-1})}. \tag{26}$$

The dimension and coefficient bounds above apply to every recursive call, so every ellipsoid query has encoding length $n^{\mathcal{O}(r)}$. For fixed ε and $\mathcal{C}$, (26) is quasipolynomial. Together with Corollary 15, this gives a deterministic QPTAS for Max Independent Set on $\mathcal{C}$.

6 Vertex weights

We reduce MWIS to MIS by rounding the vertex weights and replacing each vertex by an independent set whose size is determined by its rounded weight. We first show that this operation preserves strongly edge-sublinear separators, possibly with a larger exponent.

An *independent blow-up* of G is obtained by replacing each vertex v by a nonempty independent set B_v. For distinct u, v, all edges between B_u and B_v are present if $uv \in E(G)$, and none are present otherwise. Let $\mathcal{C}^{\mathrm{bl}}$ be the class of independent blow-ups of graphs in $\mathcal{C}$.

▶ **Lemma 16.** *Suppose $\mathcal{C}$ is hereditary and satisfies* (2)*. Then $\mathcal{C}^{\mathrm{bl}}$ is hereditary, and every m-edge graph $G' \in \mathcal{C}^{\mathrm{bl}}$ satisfies*

$$\mathrm{tw}(G') = \mathcal{O}_{\mathcal{C}}\big(m^{1-1/(2a)}\big).$$

Consequently, $\mathcal{C}^{\mathrm{bl}}$ has strongly edge-sublinear separators.

Proof. The class $\mathcal{C}^{\mathrm{bl}}$ is hereditary: an induced subgraph of an independent blow-up of G is an independent blow-up of $G[U]$, where U consists of the vertices of G with at least one surviving copy.

Let G' be an independent blow-up of $G \in \mathcal{C}$, and assume $m := |E(G')| \geqslant 1$. For an integer $s \geqslant 1$, set

$$S := \{v \in V(G') : |N_{G'}(v)| > s\}.$$

By the degree sum formula, $|S| \leqslant 2m/s$. Consider a set B_v with more than s vertices. Every neighbor of B_v has degree at least $|B_v| > s$, and hence belongs to S. Thus all vertices of $B_v \setminus S$ are isolated in $G' - S$.

It follows that, apart from isolated vertices, $G' - S$ is obtained from G by replacing each vertex by a possibly empty independent set of at most s vertices. A tree-decomposition of G can therefore be transformed into one of this graph by replacing each vertex in every bag by its copies that are nonisolated in $G' - S$. The bag sizes increase by a factor of at most s. Adding the isolated vertices in singleton bags, and then adding S to every bag, gives

$$\operatorname{tw}(G') + 1 \leqslant \frac{2m}{s} + s\big(\operatorname{tw}(G) + 1\big).$$

Since $|E(G)| \leqslant m$, (2) gives $\operatorname{tw}(G) + 1 = \mathcal{O}_{\mathcal{C}}(m^{1-1/a})$. Choosing $s := \lceil m^{1/(2a)} \rceil$ yields

$$\operatorname{tw}(G') + 1 = \mathcal{O}_{\mathcal{C}}\left(\frac{m}{s} + s m^{1-1/a}\right) = \mathcal{O}_{\mathcal{C}}\big(m^{1-1/(2a)}\big).$$

For $m = 0$, the graph G' is edgeless. Finally, a tree-decomposition of width w gives a balanced separator of size at most $w + 1$. ◀

▶ **Theorem 2.** *Let $\mathcal{C}$ have strongly edge-sublinear separators. Then* Max Weight Independent Set *admits a QPTAS on $\mathcal{C}$.*

Proof. Let $G \in \mathcal{C}$ have n vertices with nonnegative rational weights w_v, and fix rational $0 < \varepsilon < 1$. The cases $n \leqslant 1$ and $w_v = 0$ for all v are immediate. Otherwise, put

$$w_{\max} := \max_{v \in V(G)} w_v, \qquad \tau := \frac{\varepsilon w_{\max}}{2n}, \qquad t_v := \left\lfloor \frac{w_v}{\tau} \right\rfloor.$$

Construct G' by replacing each vertex v by an independent set of size t_v, omitting vertices with $t_v = 0$. Then

$$|V(G')| = \sum_{v \in V(G)} t_v \leqslant \sum_{v \in V(G)} \frac{w_v}{\tau} \leqslant \frac{n w_{\max}}{\tau} = \frac{2n^2}{\varepsilon}.$$

Since $\mathcal{C}$ is hereditary, $G' \in \mathcal{C}^{\mathrm{bl}}$. By Lemma 16, we may apply the unweighted QPTAS from Section 5 to G' with accuracy $\varepsilon/2$, obtaining an independent set I'. Return the set I of vertices of G having at least one copy in I'; I is an independent set.

Write $w(J) := \sum_{v \in J} w_v$, and let I^* be an optimum weighted independent set of G, of weight $\alpha_w(G) := w(I^*)$. Since all copies of the vertices in I^* form an independent set in G', we have

$$\tau \alpha(G') \geqslant \tau \sum_{v \in I^*} t_v \geqslant \sum_{v \in I^*} (w_v - \tau) = \alpha_w(G) - |I^*|\tau \geqslant \alpha_w(G) - n\tau \geqslant (1 - \varepsilon/2)\alpha_w(G),$$

where the last inequality uses $n\tau = \varepsilon w_{\max}/2 \leqslant \varepsilon \alpha_w(G)/2$. Since $w_v \geqslant \tau t_v$ and I' contains at most t_v copies of each $v \in I$, we have $w(I) \geqslant \tau |I'|$. Thus

$$w(I) \geqslant (1 - \varepsilon/2)\tau \alpha(G') \geqslant (1 - \varepsilon/2)^2 \alpha_w(G) \geqslant (1 - \varepsilon)\alpha_w(G).$$

The integers t_v are computed by exact rational arithmetic, in time polynomial in the binary encoding length of the weights and ε. The number of copies is polynomial in n and $1/\varepsilon$. By Lemma 16, the parameter a in (26) is replaced by $2a$ for $\mathcal{C}^{\mathrm{bl}}$. Consequently, the unweighted call takes time

$$2^{\mathcal{O}_{\mathcal{C}}\left(\varepsilon^{-(8a-2)}(\log n + \log \varepsilon^{-1})^{16a-1}\right)}.$$

Since $n \geqslant 2$, this is $2^{(\varepsilon^{-1} \log n)^{\mathcal{O}_{\mathcal{C}}(1)}}$. Including the polynomial work on the original weights gives the stated running time. ◀

AI disclosure. GPT-6 Pro first found a randomized QPTAS for MIS in any H-induced-minor-free class. The initial argument relied on the exponent not exceeding 1/2 in the balanced separators of size $\widetilde{O}(m^{1/2})$ for m-edge graphs. We thought that this should only be an artifact of the proof, and suggested the new notion of strongly edge-sublinear separators. Our intuition turned out to be correct. Making the algorithm deterministic and able to handle weights was relatively simple. The authors rewrote the eventual proof.

This is a typical example where one needs to be well familiar with a selection of techniques, here: (1) the Leighton–Rao multicommodity flow–cut theorem and its use in the context of excluded induced minors following Korhonen and Lokshtanov [39], (2) conditioning and entropy-decrease arguments, (3) the round-or-cut approach.
An autoformalized Lean proof can be found on Lax: `https://laxarchive.org/lax-263307`.

References

1 Anna Adamaszek, Sariel Har-Peled, and Andreas Wiese. Approximation schemes for independent set and sparse subsets of polygons. *Journal of the ACM*, 66(4):29:1–29:40, 2019. doi:10.1145/3326122.

2 Anna Adamaszek and Andreas Wiese. Approximation schemes for maximum weight independent set of rectangles. In *54th Annual IEEE Symposium on Foundations of Computer Science (FOCS 2013)*, pages 400–409. IEEE, 2013. doi:10.1109/FOCS.2013.50.

3 Stefan Arnborg and Andrzej Proskurowski. Linear time algorithms for NP-hard problems restricted to partial k-trees. *Discrete Applied Mathematics*, 23(1):11–24, 1989. doi:10.1016/0166-218X(89)90031-0.

4 Brenda S. Baker. Approximation algorithms for NP-complete problems on planar graphs. *J. ACM*, 41(1):153–180, 1994. doi:10.1145/174644.174650.

5 Boaz Barak, Prasad Raghavendra, and David Steurer. Rounding semidefinite programming hierarchies via global correlation. In *52nd Annual IEEE Symposium on Foundations of Computer Science (FOCS 2011)*, pages 472–481. IEEE, 2011. doi:10.1109/FOCS.2011.95.

6 Punyashloka Biswal, James R. Lee, and Satish Rao. Eigenvalue bounds, spectral partitioning, and metrical deformations via flows. *Journal of the ACM*, 57(3):13:1–13:23, 2010. arXiv:0808.0148, doi:10.1145/1706591.1706593.

7 Édouard Bonnet and Yeonsu Chang. Max independent set remains NP-hard when excluding a planar induced minor, 2026. arXiv:2609.11285.

8 Édouard Bonnet, Jadwiga Czyżewska, Tomáš Masařík, Marcin Pilipczuk, and Paweł Rzążewski. QPTAS for MWIS and finding large sparse induced subgraphs in graphs with few independent long holes. In *20th Scandinavian Symposium on Algorithm Theory (SWAT 2026)*, volume 370 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 9:1–9:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2026. doi:10.4230/LIPIcs.SWAT.2026.9.

9 Édouard Bonnet and Robert Hickingbotham. Induced minors and region intersection graphs. *Innovations in Graph Theory*, 2:313–327, 2025. doi:10.5802/igt.15.

10 Édouard Bonnet, Hung Le, Marcin Pilipczuk, and Michał Pilipczuk. Coarse balanced separators in fat-minor-free graphs, 2026. arXiv:2604.11318.

11 Robert D. Carr, Lisa K. Fleischer, Vitus J. Leung, and Cynthia A. Phillips. Strengthening integrality gaps for capacitated network design and covering problems. In *Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 106–115. SIAM, 2000. doi:10.1145/338219.338241.

12 Deeparnab Chakrabarty and Maryam Negahbani. Generalized center problems with outliers. In *45th International Colloquium on Automata, Languages, and Programming (ICALP 2018)*, volume 107 of *LIPIcs*, pages 30:1–30:14. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2018. doi:10.4230/LIPIcs.ICALP.2018.30.

13 Timothy M. Chan and Sariel Har-Peled. Approximation algorithms for maximum independent set of pseudo-disks. *Discrete & Computational Geometry*, 48:373–392, 2012. doi:10.1007/s00454-012-9417-5.

14 Miroslav Chlebík and Janka Chlebíková. The complexity of combinatorial optimization problems on d-dimensional boxes. *SIAM Journal on Discrete Mathematics*, 21(1):158–169, 2007. doi:10.1137/050629276.

15 Maria Chudnovsky, J. Pascal Gollin, Matjaž Krnc, and Martin Milanič. Dominated balanced separators in wheel-induced-minor-free graphs, 2025. arXiv:2512.12329.

16 Maria Chudnovsky, Marcin Pilipczuk, Michał Pilipczuk, and Stéphan Thomassé. Quasi-polynomial time approximation schemes for the maximum weight independent set problem in H-free graphs. *SIAM Journal on Computing*, 53(1):47–86, 2024. arXiv:1907.04585, doi:10.1137/20M1333778.

17 Bruno Courcelle, Johann A. Makowsky, and Udi Rotics. Linear time solvable optimization problems on graphs of bounded clique-width. *Theory of Computing Systems*, 33(2):125–150, 2000. doi:10.1007/s002249910009.

18 Jadwiga Czyżewska, Tomáš Masařík, Marcin Pilipczuk, Amadeus Reinald, and Paweł Rzążewski. Induced Erdős–Pósa property for long holes, long thetas, and beyond, 2026. arXiv:2607.07697.

19 Clément Dallard, Martin Milanič, and Kenny Štorgel. Treewidth versus clique number. III. Tree-independence number of graphs with a forbidden structure. *Journal of Combinatorial Theory, Series B*, 167:338–391, 2024. arXiv:2206.15092, doi:10.1016/j.jctb.2024.03.005.

20 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing (STOC 2014)*, pages 624–633. ACM, 2014. doi:10.1145/2591796.2591884.

21 Zdeněk Dvořák. Sublinear separators, fragility and subexponential expansion. *European Journal of Combinatorics*, 52:103–119, 2016. doi:10.1016/j.ejc.2015.09.001.

22 Zdeněk Dvořák. On classes of graphs with strongly sublinear separators. *European Journal of Combinatorics*, 71:1–11, 2018. doi:10.1016/j.ejc.2018.02.032.

23 Zdeněk Dvořák and Abhiruk Lahiri. Approximation schemes for bounded distance problems on fractionally treewidth-fragile graphs. In *29th Annual European Symposium on Algorithms (ESA 2021)*, volume 204 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 40:1–40:10. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2021. doi:10.4230/LIPIcs.ESA.2021.40.

24 Zdeněk Dvořák and Sergey Norin. Treewidth of graphs with balanced separations. *Journal of Combinatorial Theory, Series B*, 137:137–144, 2019. doi:10.1016/j.jctb.2018.12.007.

25 Thomas Erlebach, Klaus Jansen, and Eike Seidel. Polynomial-time approximation schemes for geometric intersection graphs. *SIAM Journal on Computing*, 34(6):1302–1323, 2005. doi:10.1137/S0097539702402676.

26 Peter Gartland. *Quasi-Polynomial Time Techniques for Independent Set and Beyond in Hereditary Graph Classes*. PhD thesis, University of California, Santa Barbara, 2023. URL: https://www.escholarship.org/uc/item/0kk6d2jv.

27 Peter Gartland and Daniel Lokshtanov. Independent set on P_k-free graphs in quasi-polynomial time. In *2020 IEEE 61st Annual Symposium on Foundations of Computer Science (FOCS)*, pages 613–624. IEEE, 2020. doi:10.1109/FOCS46700.2020.00063.

28 Peter Gartland, Daniel Lokshtanov, Marcin Pilipczuk, Michał Pilipczuk, and Paweł Rzążewski. Finding large induced sparse subgraphs in $C_{>t}$-free graphs in quasipolynomial time. In *STOC*, 2021. arXiv:2007.11402v3, doi:10.1145/3406325.3451034.

29 Fănică Gavril. Algorithms for a maximum clique and a maximum independent set of a circle graph. *Networks*, 3(3):261–273, 1973. doi:10.1002/net.3230030305.

30 Martin Grohe. Local tree-width, excluded minors, and approximation algorithms. *Combinatorica*, 23(4):613–632, 2003. doi:10.1007/s00493-003-0037-9.

31 Martin Grohe and Dániel Marx. On tree width, bramble size, and expansion. *Journal of Combinatorial Theory, Series B*, 99(1):218–228, 2009. doi:10.1016/j.jctb.2008.06.004.

32 Martin Grötschel, László Lovász, and Alexander Schrijver. *Geometric Algorithms and Combinatorial Optimization*, volume 2 of *Algorithms and Combinatorics*. Springer, second edition, 1993. doi:10.1007/978-3-642-78240-4.

33 Andrzej Grzesik, Tereza Klimošová, Marcin Pilipczuk, and Michał Pilipczuk. Polynomial-time algorithm for maximum weight independent set on P_6-free graphs. *ACM Trans. Algorithms*, 18(1):4:1–4:57, 2022. doi:10.1145/3414473.

34 Sariel Har-Peled. Approximately: Independence implies vertex cover, 2023. arXiv:2308.00840.

35 Sariel Har-Peled and Kent Quanrud. Approximation algorithms for polynomial-expansion and low-density graphs. *SIAM Journal on Computing*, 46(6):1712–1744, 2017. doi:10.1137/16M1079336.

36 Johan Håstad. Clique is hard to approximate within $n^{1-\varepsilon}$. *Acta Mathematica*, 182(1):105–142, 1999. doi:10.1007/BF02392825.

37 Shlomo Hoory, Nathan Linial, and Avi Wigderson. Expander graphs and their applications. *Bulletin of the American Mathematical Society*, 43(4):439–561, 2006. doi:10.1090/S0273-0979-06-01126-8.

38 Richard M. Karp. Reducibility among combinatorial problems. The IBM Research Symposia Series, pages 85–103. Plenum Press, New York, 1972. URL: https://doi.org/10.1007/978-1-4684-2001-2_9, doi:10.1007/978-1-4684-2001-2_9.

39 Tuukka Korhonen and Daniel Lokshtanov. Induced-minor-free graphs: Separator theorem, subexponential algorithms, and improved hardness of recognition. In David P. Woodruff, editor, *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2024)*, pages 5249–5275. SIAM, 2024. arXiv:2308.04795, doi:10.1137/1.9781611977912.188.

40 James R. Lee. Separators in region intersection graphs. In Christos H. Papadimitriou, editor, *8th Innovations in Theoretical Computer Science Conference, ITCS 2017, January 9-11, 2017, Berkeley, CA, USA*, volume 67 of *LIPIcs*, pages 1:1–1:8. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2017. URL: https://doi.org/10.4230/LIPIcs.ITCS.2017.1, doi:10.4230/LIPICS.ITCS.2017.1.

41 Tom Leighton and Satish Rao. Multicommodity max-flow min-cut theorems and their use in designing approximation algorithms. *Journal of the ACM*, 46(6):787–832, 1999. doi:10.1145/331524.331526.

42 Jiří Matoušek. Near-optimal separators in string graphs. *Combinatorics, Probability and Computing*, 23(1):135–139, 2014. doi:10.1017/S0963548313000400.

43 Amir Nikabadi and Paweł Rzążewski. Induced packing treewidth, 2026. arXiv:2607.07595.

44 Sang-il Oum and Paul D. Seymour. Approximating clique-width and branch-width. *Journal of Combinatorial Theory, Series B*, 96(4):514–528, 2006. doi:10.1016/j.jctb.2005.10.006.

45 Michał Pilipczuk, Erik Jan van Leeuwen, and Andreas Wiese. Quasi-polynomial time approximation schemes for packing and covering problems in planar graphs. *Algorithmica*, 82(6):1703–1739, 2020. doi:10.1007/s00453-019-00670-w.

46 Prasad Raghavendra and Ning Tan. Approximating CSPs with global cardinality constraints using SDP hierarchies. In *Proceedings of the Twenty-Third Annual ACM-SIAM Symposium on Discrete Algorithms (SODA 2012)*, pages 373–387. SIAM, 2012. doi:10.1137/1.9781611973099.33.

47 Neil Robertson and Paul D. Seymour. Graph minors. II. Algorithmic aspects of tree-width. *Journal of Algorithms*, 7(3):309–322, 1986. doi:10.1016/0196-6774(86)90023-4.

48 Hanif D. Sherali and Warren P. Adams. A hierarchy of relaxations between the continuous and convex hull representations for zero-one programming problems. *SIAM Journal on Discrete Mathematics*, 3(3):411–430, 1990. doi:10.1137/0403036.

49 David Zuckerman. Linear degree extractors and the inapproximability of Max Clique and Chromatic Number. *Theory of Computing*, 3(6):103–128, 2007. doi:10.4086/toc.2007.v003a006.