Bi-Criteria Approximations for Vertex Deletion Problems and d-Hitting Set

Soumen Mandal
Department of Mathematics, IIT Delhi, New Delhi, India

Ashutosh Rai
Department of Mathematics, IIT Delhi, New Delhi, India

Saket Saurabh
Institute of Mathematical Sciences, HBNI, Chennai, India
University of Bergen, Norway

Abstract

We study bi-criteria approximation algorithms for vertex deletion problems in the (k, W) setting, where both the solution size and total weight are bounded simultaneously. Given a graph G, a weight function $w : V \to \mathbb{Q}^+$, a size bound k, and a weight budget W, a bi-criteria (a, b)-approximation algorithm either certifies that no solution of size at most k and weight at most W exists, or returns a solution of size at most ak and weight at most bW. Parameterizing by the solution size k – rather than the weight budget W – allows our algorithms to handle arbitrary positive rational weights without any lower bound assumption, addressing a fundamental limitation of prior W-parameterized approaches.

We obtain two families of results. For general vertex deletion problems Π-Deletion admitting a polynomial-time weighted α-approximation, we obtain a polynomial-time $\left(\alpha(\lambda+1), \alpha\left(1+\frac{1}{\lambda}\right)\right)$-approximation for any $\lambda > 0$, a randomized FPT improvement for problems admitting a *sampling step*, and a deterministic FPT version for problems with bounded obstruction size. For (k, W)-d-Hitting Set, which captures vertex deletion problems with obstruction size at most d, we design a polynomial-time (d, d)-approximation, a parameterized family of $((1-\epsilon)d,\, d)$-approximations improving the size factor below d, and two algorithms that simultaneously push both factors below d: a $\left(\frac{d+1}{2}, \frac{d+1}{2}\right)$-approximation and a more refined $(d-\gamma, d-\gamma)$-approximation for any $\gamma \in \left(0, \frac{d-1}{2}\right)$. All algorithms work with arbitrary positive rational weights and are parameterized by the solution size k.

To demonstrate the broad applicability of our framework, we instantiate our results on six well-studied vertex deletion problems: Cluster Vertex Deletion, FVS in Tournaments, Split Vertex Deletion, Feedback Vertex Set, d-Path Vertex Cover, and Pathwidth-One Vertex Deletion. In fact, our general results apply to any vertex deletion problem admitting a polynomial-time weighted approximation algorithm, and the six problems serve as representative examples spanning a range of obstruction structures – from bounded-size obstructions to unbounded ones. For (k, W) setting of Feedback Vertex Set and Pathwidth-One Vertex Deletion, we establish new sampling steps enabling the FPT approximation results. For Pathwidth-One Vertex Deletion, we additionally prove a polynomial-time 3-approximation for the weighted version on general graphs.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms

Keywords and phrases Parameterized approximation algorithms, bi-criteria approximation, vertex deletion problems, d-Hitting Set, branching algorithms, sampling step

Digital Object Identifier 10.4230/LIPIcs.MFCS.2026.15

Funding *Soumen Mandal*: Supported by IITD Seed Grant (MI03046GSG).

Acknowledgements The authors thank Ishan Chakraborty and Madhumita Kundu for valuable discussions during the initial stages of this work.

51st International Symposium on Mathematical Foundations of Computer Science (MFCS 2026).
Editors: Michal Koucký and Daniela Petrişan; Article No. 15; pp. 15:1–15:18
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Vertex deletion problems form a fundamental theme in graph theory and algorithm design. Given a graph $G = (V, E)$ and a graph property Π, the Π-Deletion problem asks for a minimum-size set $S \subseteq V$ whose removal leaves a graph satisfying Π – that is, $G - S$ satisfies Π. By choosing Π appropriately, one can enforce diverse structural constraints – such as being an *independent set*, *acyclic*, or a collection of *clusters* – making these problems highly relevant both theoretically and in practice. Well-known examples include Vertex Cover (where Π is the property of being an independent set), Feedback Vertex Set (where Π is acyclicity), and Cluster Vertex Deletion (where Π is being a cluster graph). These problems arise naturally in diverse application domains: in *network reliability*, faulty nodes are removed to preserve connectivity; in *data clustering*, outliers are eliminated to improve cluster quality; and in *computational biology*, genes or proteins are removed from interaction networks to study specific functional modules.

Often, vertices differ in importance or cost of removal, leading to the *weighted* variant, where each vertex carries a non-negative weight and the goal is to find a minimum-weight deletion set achieving the target property. Both variants have been extensively studied. A seminal result of Lewis and Yannakakis [21] shows that many vertex deletion problems are NP-hard, motivating the study of *near optimal solutions* that is, *approximation algorithms*. The best-known polynomial-time approximation ratio for Vertex Cover, Feedback Vertex Set, and Cluster Vertex Deletion in the weighted setting is 2 [24, 3, 2], and this is essentially tight: under the *Unique Games Conjecture*, no polynomial-time $(2 - \epsilon)$-approximation exists for any $\epsilon > 0$ [18].

While approximation algorithms aim to compute near-optimal solutions efficiently for large inputs, another prominent paradigm, called *parameterized algorithms*, takes a different approach by focusing on a specific parameter. In vertex deletion problems, this parameter is often the solution size k, i.e., the number of vertices to delete. The goal is to design algorithms whose running time remains efficient when k is small, even if the graph itself is large, typically of the form $f(k) \cdot n^{\mathcal{O}(1)}$ for some computable function f. This gives rise to the framework of *fixed-parameter tractability* (FPT), which has produced powerful exact algorithms for many vertex deletion problems. When parameterized by the solution size, problems such as Vertex Cover, Feedback Vertex Set, and Cluster Vertex Deletion admit FPT algorithms. The best known FPT algorithms for these problems run in $\mathcal{O}^*(1.25284^k)$[1] [13], $\mathcal{O}^*(3.46^k)$ [14], and $\mathcal{O}^*(1.811^k)$ [28], respectively.

A related line of research, called *parameterized approximation algorithms*, serves as a bridge between parameterized exact algorithms and approximation algorithms. These algorithms provide better approximation ratios than those achievable by polynomial-time algorithms – sometimes even circumventing known hardness conjectures – while running faster than exact parameterized algorithms, with running time of the form $f(k) \cdot n^{\mathcal{O}(1)}$ for a given parameter k. Early work by Fellows et al. [11] introduced the framework of *fidelity preserving transformations* for designing parameterized approximation algorithms. A comprehensive survey of hardness and algorithmic results in this area is given by Feldmann et al. [10]. For Vertex Cover, Brankovic and Fernau [4] were among the first to obtain a parameterized α-approximation for $\alpha < 2$ in FPT time, circumventing the polynomial-time hardness barrier. Subsequently, Kulik and Shachnai [20] introduced *randomized branching* and substantially improved the running times. For Feedback Vertex Set, Jana et al. [15]

[1] We use the $\mathcal{O}^*$ notation to suppress factors polynomial in the input size n, so $\mathcal{O}^*(f(k)) = f(k) \cdot n^{\mathcal{O}(1)}$.

recently designed a $(1+\epsilon)$-approximation scheme for every $\epsilon \in (0,1)$, achieving a better running time than the best known FPT algorithm for every $\epsilon > 0$. Very recently, a generic "sampling with a black box" technique was introduced in [9] for designing parameterized approximation algorithms for vertex deletion problems including Feedback Vertex Set, d-Hitting Set, and ℓ-Path Vertex Cover, by combining a randomized sampling step with either an exact FPT algorithm or a polynomial-time approximation. We refer the reader to the survey of [10] for a comprehensive overview of the area. The (k,W)-Setting and the Need for Simultaneous Bounds.

One limitation of most parameterized approximation algorithms is that they primarily address the *unweighted* setting. On the other hand, algorithms for the *weighted* version of vertex deletion problems typically have no control over the *size* of the returned solution, which can be arbitrarily large. This creates a fundamental tension: approximation algorithms control the weight but not the cardinality, while parameterized algorithms control the cardinality but are often restricted to unweighted instances.

To understand why controlling *both* size and weight simultaneously is important, consider the following motivating scenarios. In a network where vertices represent servers and weights represent operational costs, one may wish to remove a feedback vertex set subject to both a budget constraint (total cost $\leq W$) *and* a manpower constraint (number of servers $\leq k$). Neither a pure approximation algorithm (which may return an arbitrarily large number of servers) nor a pure parameterized algorithm (which ignores operational costs) addresses this requirement. Similarly, in computational biology, one may wish to remove a bounded number of genes (size constraint) with bounded total disruption (weight constraint) to achieve a desired interaction network structure.

The study of weighted vertex deletion problems from a parameterized perspective has primarily focused on W as the parameter. Early work of Niedermeier and Rossmanith [25] parameterized weighted Vertex Cover by W, and Shachnai and Zehavi [27] later improved such running times from $\mathcal{O}^*(c^W)$ to $\mathcal{O}^*(c^s)$ where $s \leq W$ is the minimum solution size, via a general multivariate framework covering Vertex Cover, 3-Hitting Set, and Edge Dominating Set. However, parameterizing by W carries two inherent limitations. First, all such algorithms require vertex weights to be at least 1; with arbitrary positive rational weights, the problem is no longer FPT parameterized by W under standard complexity assumptions [23]. Second, W can be arbitrarily large – potentially exponential in the input size – making it a less natural parameter than the solution size k for measuring instance difficulty.

The (k,W)-Π-Deletion setting addresses both of these concerns. By parameterizing by the solution size k and treating W as a hard constraint rather than a parameter, the running times of our algorithms depend only on k. No lower bound on the weights is assumed: the weight function $w : V \to \mathbb{Q}^+$ may assign arbitrary positive rationals to vertices, and our algorithms remain valid regardless of the magnitude of W. This makes the (k,W) framework a natural complement to W-parameterized algorithms.

This setting has been considered in [6, 12, 19, 23]. For example, when parameterized by k, the best known exact parameterized algorithms for (k,W)-Feedback Vertex Set and (k,W)-Vertex Cover run in $\mathcal{O}^*(3.619^k)$ [1] and $\mathcal{O}^*(1.4656^k)$ [23], respectively. When the maximum size of a minimal obstruction for Π is bounded by a constant d, any instance of (k,W)-Π-Deletion can be naturally formulated as an instance of (k,W)-d-Hitting Set, a setting we exploit extensively in this work.

(k, W)-Π-DELETION
Input: A graph $G = (V, E)$, a weight function $w : V \to \mathbb{Q}^+$, a positive integer k, and a positive number W.
Question: Does there exist a set $S \subseteq V$ of size at most k and weight at most W such that $G - S$ satisfies the graph property Π?

A natural algorithmic goal in this setting is to design *bi-criteria approximation algorithms*: algorithms that, given a (k, W)-Π-DELETION instance, either correctly determine that no solution of size at most k and weight at most W exists, or return a solution of size at most ak and weight at most bW for factors $a, b \geq 1$. The pair (a, b) is the *bi-criteria approximation factor*. The challenge is to simultaneously control both parameters, which requires fundamentally different techniques from either pure approximation or pure parameterized algorithms.

In a recent work [23], bi-criteria approximation was studied for (k, W)-VERTEX COVER. Results included a polynomial-time $(2, 2)$-approximation, a parameterized $(1 + \epsilon, 2)$-approximation, and a parameterized $(2 - \delta, 2 - \delta)$-approximation, all achieving constant-factor approximations on both size and weight simultaneously. In this paper, we extend this line of investigation to a broad class of vertex deletion minimization problems, going significantly beyond VERTEX COVER.

Our Results and Methods. Vertex deletion minimization problems can be broadly classified into two categories: those whose obstruction set has bounded size, and those without. By an *obstruction*, we mean a subgraph that is not allowed in the remaining graph. Instances where the maximum size of a minimal obstruction is bounded by d can be naturally formulated as instances of the d-HITTING SET problem. We address both categories, beginning with the more general setting.

Results for General Π-DELETION Problems (Section 3).minimization problems where Π-DELETION admits a polynomial-time α-approximation for its weighted variant. The central difficulty in the (k, W)-Π-DELETION setting is that existing weighted approximation algorithms guarantee only a weight bound on the solution, with no control over its size, while existing parameterized algorithms control size but ignore weights entirely. We reconcile these two goals.

Our first result is a polynomial-time $\left(\alpha(\lambda + 1),\, \alpha\left(1 + \frac{1}{\lambda}\right)\right)$-approximation for (k, W)-Π-DELETION, for any $\lambda > 0$. The key idea is to construct a modified weight function $w'(v) = w(v) + \frac{W}{\lambda k}$ that embeds the size constraint into the weight, and then run the known α-approximation on this augmented instance. Remarkably, the solution returned simultaneously satisfies both the weight bound $\alpha(1 + \frac{1}{\lambda})W$ and the size bound $\alpha(\lambda + 1)k$, whenever a solution of size at most k and weight at most W exists. The parameter λ controls the trade-off between the size and weight approximation factors.

We then improve the size approximation factor in FPT time for problems admitting a *sampling step* – a randomized polynomial-time procedure that, given a (k, W)-Π-DELETION instance with a feasible solution, returns a vertex belonging to some optimal solution with probability at least $q \in (0, 1)$. We show that such a sampling step exists for several important problems, including (k, W)-FEEDBACK VERTEX SET (with $q = \frac{1}{8}$, proved here) and (k, W)-PATHWIDTH-ONE VERTEX DELETION (with $q = \frac{1}{7}$, proved here). Our algorithm iteratively applies the sampling step to peel off a set S_1 of approximately ϵk vertices from an optimal solution, and then invokes the polynomial-time α-approximation on the reduced instance. This yields, for any $\lambda > 0$ and $\epsilon \in \left[0, \lambda + 1 - \frac{1}{\alpha}\right]$, a randomized $\left(\alpha(\lambda + 1 - \epsilon),\, \alpha\left(1 + \frac{1}{\lambda}\right)\right)$-

approximation with running time $\mathcal{O}^*\left(q^{-\frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}}\right)$. The running time analysis relies on a careful calculation of how many sampling steps are needed before the residual instance falls within range of the polynomial-time approximation.

Finally, when Π-Deletion has bounded obstruction size c, we derandomize the sampling step by replacing it with a deterministic branching over all at most c vertices of an obstruction. The branching stops once the size parameter drops below a threshold $x = \frac{(\alpha(\lambda+1-\epsilon)-1)k}{\alpha(\lambda+1)-1}$, at which point the polynomial-time α-approximation is applied to the leaf instance. Writing the solution as $S = S_1 \uplus S_2$ where S_1 are the branching vertices and S_2 is the leaf solution, the size bound $|S| \leq \alpha(\lambda + 1 - \epsilon)k$ follows from the choice of x, and the weight bound $w(S) \leq \alpha(1 + \frac{1}{\lambda})W$ follows since $w(S_2) \leq \alpha(1 + \frac{1}{\lambda})W' \leq \alpha(1 + \frac{1}{\lambda})W$. The resulting deterministic algorithm runs in time $\mathcal{O}^*\left(c^{\frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}}\right)$.

Results for (k, W)-d-Hitting Set (Section 4).size d, the d-Hitting Set formulation allows us to go further and obtain results that simultaneously improve *both* approximation factors below d, which the general framework cannot achieve on its own.

> (k, W)-d-Hitting Set ((k, W)-dHS)
> **Input:** A universe $\mathcal{U}$, a family $\mathcal{H}$ of subsets of $\mathcal{U}$ each of size at most d, a weight function $w : \mathcal{U} \to \mathbb{Q}^+$, a positive integer k, and a positive number W.
> **Question:** Does there exist a set $S \subseteq \mathcal{U}$ of size at most k and weight at most W such that $S \cap H \neq \emptyset$ for all $H \in \mathcal{H}$?

We begin with a polynomial-time (d, d)-approximation Poly_apprx_dHS for (k, W)-dHS, obtained by solving the natural LP relaxation and rounding: any element e with $x_e^* \geq \frac{1}{d}$ is included in the solution. The integrality gap of the LP gives the (d, d) guarantee, generalizing the $(2, 2)$-approximation for (k, W)-Vertex Cover [23].

To improve the size factor below d while keeping the weight factor at d, we use a branching algorithm that repeatedly selects an unhit set $H \in \mathcal{H}$ and branches on each of its d elements, including one in the solution and recursing. Branching stops when the size parameter drops below the threshold $x = \frac{(1-\epsilon)d-1}{d-1}k$, at which point the (d, d)-approximation is applied. The measure $\mu = k - x$ drops by 1 at each branching step, giving branching number d and total running time $\mathcal{O}^*\left(d^{\frac{\epsilon d}{d-1} \cdot k}\right)$. The size analysis writes the output as $S = S_1 \uplus S_2$: the branching vertices satisfy $|S_1| = k - k'$, and the leaf solution satisfies $|S_2| \leq dk'$, giving $|S| \leq k + (d-1)k' \leq (1-\epsilon)dk$.

To simultaneously push *both* factors below d, we introduce a more sophisticated $\left(1 + (d-1)^2\right)$-way branching. The algorithm first applies two reduction rules: removing singleton sets (whose unique element must be in the solution) and removing dominated sets (where a maximum-weight element can be assumed to be included). On the remaining instance, we branch on a maximum-weight element u appearing in at least two sets via two types of branches: *Branch-1* includes u directly; *Branch-2* (pair-branches) selects one element from each of two sets H_1, H_2 containing u but not u itself, giving $(d-1)^2$ pair-branch children. Branching continues until depth $\frac{k}{2}$, where the (d, d)-approximation handles the residual. This yields a $\left(\frac{d+1}{2}, \frac{d+1}{2}\right)$-approximation running in time $\mathcal{O}^*\left((1 + (d-1)^2)^{\frac{k}{2}}\right)$, and specializes to the $(1.5, 1.5)$-approximation for (k, W)-Vertex Cover [23] when $d = 2$. Under the Unique Games Conjecture, d-Hitting Set is NP-hard to approximate within a factor of $d - \epsilon$ for any constant $\epsilon > 0$ in polynomial time [18], making the factor d essentially tight. This makes our FPT results that push both approximation factors simultaneously below d particularly meaningful.

Finally, we obtain a $(d-\gamma, d-\gamma)$-approximation for any $\gamma \in \left(0, \frac{d-1}{2}\right)$, with running time $\mathcal{O}^*(M_{\gamma,d}^k)$, where $M_{\gamma,d} \geq 1$ is a constant determined by γ and d via an optimization over a closed-form expression (see equation (1)). This provides a parameterized family of algorithms interpolating between the $\left(\frac{d+1}{2}, \frac{d+1}{2}\right)$-approximation (as $\gamma \to \frac{d-1}{2}$) and the (d,d)-approximation (as $\gamma \to 0$), though the running time $M_{\gamma,d}^k$ does not decrease as smoothly with γ as one might hope, particularly for $d \geq 3$. The algorithm uses the same $\left(1 + (d-1)^2\right)$-way branching with two global thresholds: a size threshold $x = \frac{d-\gamma-1}{d-1}k$ and a weight threshold $Y = \frac{d-\gamma-1}{d-1}W$. Branching stops and Poly_apprx_dHS is invoked when either (i) both budgets have dropped below their thresholds (Type-I), or (ii) the size budget is below x and no high-weight element remains (Type-II). Branching continues on Type-IV nodes ($k' > x$) and, crucially, also on Type-III nodes ($k' \leq x$, $W' > Y$, but a high-weight element still exists). The dominant contribution to the running time comes from Type-III nodes, which can persist up to depth $\frac{k}{2}$, producing exponentially many paths for $d \geq 3$. The count of such nodes involves a combinatorial analysis yielding $\mathcal{O}^*\left(\binom{\ell}{A}(d-1)^{2(\ell-A)}\right)$ nodes at layer ℓ with A Branch-1 steps, and the maximum over all layers gives $M_{\gamma,d}$. We note that for $d = 2$, $(d-1)^2 = 1$ so the pair-branch factor vanishes and the analysis is tight. For $d \geq 3$, the $(d-1)^2$ pair-branches can independently remain as Type-III instances all the way to depth $\frac{k}{2}$, producing exponentially many paths; this is a genuine structural limitation rather than an artifact of analysis, as we discuss in Section 6.

Applications (Section 5). We examine six well-known problems to illustrate the applicability of our results. For any Π-Deletion admitting a weighted α-approximation, at least one of our general results applies; for problems with additional structure – constant obstruction size or a sampling step – multiple results apply simultaneously, giving a richer set of trade-offs.

For Cluster Vertex Deletion ($d = 3$, $\alpha = 2$, $c = 3$), FVS in Tournaments ($d = 3$, $\alpha = 2$, $c = 3$), and Split Vertex Deletion ($d = 5$, $\alpha = 2 + \sigma$, $c = 5$), all four d-Hitting Set results apply directly, as do the general vertex deletion results. For Feedback Vertex Set, which has unbounded obstructions (cycles), we prove that (k, W)-FVS admits a sampling step with probability at least $\frac{1}{8}$: after applying two reduction rules (removing degree-≤ 1 vertices and contracting long degree-2 paths), we sample a random endpoint of a random edge; the probability that any fixed optimal solution vertex is chosen is at least $\frac{1}{8}$. For d-Path Vertex Cover, all d-Hitting Set results apply directly for $d = 3$, while for $d = 4$ we use the known 3-approximation of [5]. For Pathwidth-One Vertex Deletion, we first prove a polynomial-time 3-approximation for the weighted version on general graphs: a 2-approximate weighted FVS [3] removes all cycles (contributing $\leq 2w(S^*)$ to the weight), and then a sophisticated tree DP solves weighted POVD exactly on the resulting forest (contributing $\leq w(S^*)$). The DP assigns each vertex a role of deleted, leaf, or spine, tracks whether the parent is spine via a binary flag, and uses a constant-capacity knapsack at each node to enforce the local caterpillar constraints; the total running time is $O(n)$. We also show that (k, W)-POVD admits a sampling step with probability $\frac{1}{7}$ by finding a T_2, K_3, or C_4 subgraph and sampling one of its at most 7 vertices. A summary of all general results is given in Table 1.

Organization. Section 2 introduces preliminary definitions and states two supporting results used throughout. Section 3 presents the three general bi-criteria approximation results for (k, W)-Π-Del, each with a proof sketch highlighting the key ideas. Section 4 develops the four results for (k, W)-d-Hitting Set: the first two are given with proof sketches, while the latter two – which are the most technically involved – include fuller proof sketches covering

Table 1 Summary of general bi-criteria approximation results. Here k is the solution size, W is the weight budget, d is the maximum size of a minimal obstruction, α is the polynomial-time approximation ratio for weighted Π-DEL, $q \in (0,1)$ is the sampling success probability, and $M_{\gamma,d}$ is defined in (1). All results give (a,b)-approximations: the returned solution has size $\le ak$ and weight $\le bW$, or certifies no solution of size $\le k$ and weight $\le W$ exists. ‡ indicates a randomized running time.

Setting	**Approx. factor** (a,b)	**Running time**	**Result**
(k,W)-Π-DEL *with* α*-approx. for weighted* Π-DEL			
General, $\lambda > 0$	$\left(\alpha(\lambda+1), \alpha\left(1+\frac{1}{\lambda}\right)\right)$	poly	Thm. 3
Sampling step prob q, $\epsilon \in \left[0, \lambda+1-\frac{1}{\alpha}\right]$	$\left(\alpha(\lambda+1-\epsilon), \alpha\left(1+\frac{1}{\lambda}\right)\right)$	$\mathcal{O}^*\left(q^{-\frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}}\right)$ ‡	Thm. 4
Obstruction size $\le c$, $\epsilon \in \left[0, \lambda+1-\frac{1}{\alpha}\right]$	$\left(\alpha(\lambda+1-\epsilon), \alpha\left(1+\frac{1}{\lambda}\right)\right)$	$\mathcal{O}^*\left(c^{\frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}}\right)$	Thm. 5
(k,W)-d-HITTING SET			
General	$(d,\, d)$	poly	Thm. 6
General, $\epsilon \in \left[0, \frac{d-1}{d}\right]$	$((1-\epsilon)d,\, d)$	$\mathcal{O}^*\left(d^{\frac{\epsilon d}{d-1}\cdot k}\right)$	Thm. 7
General	$\left(\frac{d+1}{2}, \frac{d+1}{2}\right)$	$\mathcal{O}^*\left((1+(d-1)^2)^{\frac{k}{2}}\right)$	Thm. 8
General, $\gamma \in \left(0, \frac{d-1}{2}\right)$	$(d-\gamma,\, d-\gamma)$	$\mathcal{O}^*\left(M^k_{\gamma,d}\right)$	Thm. 9

the approximation factor analysis and running time argument. Section 5 instantiates our framework on six problems; for PATHWIDTH-ONE VERTEX DELETION, a novel polynomial-time 3-approximation and a sampling step with probability $\frac{1}{7}$ are established, with the underlying exact forest algorithm deferred to the full version. Section 6 concludes with open directions. Proof details for all results marked † can be found in the full version of this paper.

2 Preliminaries

For a graph $G = (V, E)$, we denote $V(G)$, $E(G)$, $n = |V(G)|$, and $m = |E(G)|$. For $S \subseteq V(G)$, the subgraph induced on S is $G[S]$, and $G - S = G[V(G) \setminus S]$. The open neighborhood of v is $N(v) = \{u \in V : uv \in E(G)\}$ and $\deg(v) = |N(v)|$. For an instance $(\mathcal{U}, \mathcal{H}, w, k, W)$ of (k,W)-dHS, we define $A_{e_j} = \{H \in \mathcal{H} \mid e_j \in H\}$. An instance of a decision problem is called a `Yes` instance if the answer is yes, and a `No` instance otherwise.

A *sampling step* for a (k,W)-Π-DELETION instance is a polynomial-time randomized algorithm that, given an instance (G, w, k, W) with a feasible solution, returns a vertex belonging to some fixed optimal solution with probability at least q, for some constant $q \in (0,1)$.

For a graph property Π, a family of graphs $\mathcal{F}$ is called an *obstruction set* for Π if every graph satisfying Π contains no member of $\mathcal{F}$ as a subgraph (or induced subgraph). The *maximum obstruction size* is $\max\{|V(H)| : H \in \mathcal{F}\}$.

For branching algorithms, the *branching number* of a branching vector $(a_1, \ldots, a_l)$ is the unique positive root of $\sum_{i=1}^{l} x^{-a_i} = 1$. If c is the largest branching number over all branching steps and the measure is always non-negative, the branching tree has $\mathcal{O}(c^k)$ nodes, giving running time $\mathcal{O}^*(c^k)$ [7].

Deferred proofs. Due to space constraints, proofs of results marked with † are deferred to the full version of this paper.

▶ **Observation 1** (†). *If the maximum obstruction size for* Π-DEL *is a constant* c*, then* (k, W)-Π-DEL *admits a sampling step with success probability at least* $\frac{1}{c}$*.*

▶ **Lemma 2** (†). *Let* $I = (\mathcal{U}, \mathcal{H}, w, k, W)$ *be an instance of* (k, W)*-dHS and* $H \in \mathcal{H}$*. For each* $e_i \in H$*, define* $I_i = (\mathcal{U} \setminus \{e_i\}, \mathcal{H} \setminus A_{e_i}, w, k-1, W - w(e_i))$*. Then* I *is a* `No` *instance if and only if every* I_i *is a* `No` *instance.*

3 Bi-criteria Approximations for Vertex Deletion Problems

We present three bi-criteria approximation algorithms for (k, W)-Π-DEL, each requiring progressively more structure from the problem.

3.1 Polynomial-time Bi-criteria Approximation

The following result requires no structural assumption beyond the existence of a weighted α-approximation, and serves as the base algorithm invoked by all subsequent results.

▶ **Theorem 3.** *Let* Π-DEL *be a vertex deletion minimization problem. If there exists a polynomial-time* α*-approximation algorithm for the weighted version of* Π-DEL*, then for any* $\lambda > 0$*, there exists a polynomial-time* $\left(\alpha(\lambda + 1), \alpha\left(1 + \frac{1}{\lambda}\right)\right)$*-approximation algorithm for the* $(k, W) - $Π-DEL.

Proof. We begin by describing the approximation algorithm, which we name Poly_apprx_VDel. Suppose that Π-DEL admits a polynomial-time α-approximation algorithm $\mathcal{A}$ for its weighted version. Given an instance $I = (G, w, k, W)$ of (k, W)-Π-DEL and $\lambda > 0$, we define a new weight function $w' : V(G) \to \mathbb{Q}^+$ as follows:

$$w'(v) = w(v) + \frac{W}{\lambda k}, \quad \text{for all } v \in V(G).$$

We then apply $\mathcal{A}$ to the graph G with weight function w'. If the algorithm returns a solution S such that $|S| \leq \alpha(\lambda+1)k$ and $w(S) \leq \alpha\left(1 + \frac{1}{\lambda}\right) W$, we return S. Otherwise, we return `No`.

Since $\mathcal{A}$ runs in polynomial time and the transformation of weights takes linear time, the overall algorithm runs in polynomial time.

Now we prove the correctness. Suppose the input instance $I = (G, w, k, W)$ is a `Yes` instance of $(k, W) - $Π-DEL. Then there exists a solution $S' \subseteq V(G)$ such that $|S'| \leq k$ and $w(S') \leq W$. The weight of S' under w' is

$$w'(S') = \sum_{v \in S'} w(v) + \sum_{v \in S'} \frac{W}{\lambda k} = w(S') + \frac{W|S'|}{\lambda k} \leq W + \frac{W}{\lambda} = \left(1 + \frac{1}{\lambda}\right) W.$$

Since we apply the algorithm $\mathcal{A}$, an α-approximation for the weighted version, under the weight function w', the output S must satisfies

$$w'(S) \leq \alpha \cdot w'(S') \leq \alpha \left(1 + \frac{1}{\lambda}\right) W.$$

Expanding $w'(S)$ gives:

$$w'(S) = w(S) + \frac{W|S|}{\lambda k} \leq \alpha \left(1 + \frac{1}{\lambda}\right) W.$$

Since both terms are positive, the RHS will bound each term. Thus, rearranging gives the guarantees:

$$w(S) \leq \alpha\left(1+\frac{1}{\lambda}\right)W, \quad |S| \leq \alpha(\lambda+1)k.$$

This shows that when (G, w, k, W) is a `Yes` instance, then the set S returned by $\mathcal{A}$ satisfies $w(S) \leq \alpha\left(1+\frac{1}{\lambda}\right)W$, $|S| \leq \alpha(\lambda+1)k$, and thus our algorithm also returns this set S. The feasibility of S follows from the fact that S is returned by an approximation algorithm $\mathcal{A}$ of Π-Del. This completes the proof of the theorem. ◀

This result is fully general but has a limitation: the size factor $\alpha(\lambda+1)$ and weight factor $\alpha(1+\frac{1}{\lambda})$ are coupled through λ, and improving one worsens the other. To break this coupling and improve the size factor independently, we need additional structure from the problem.

3.2 Bi-criteria Approximation via a Sampling Step

When the problem additionally admits a sampling step, we can improve the size approximation factor below $\alpha(\lambda+1)$ at the cost of FPT running time.

▶ **Theorem 4** (†). *Let Π-Del admit a polynomial-time α-approximation and a sampling step with probability $q \in (0,1)$ for its (k,W)-variant. Then for any $\lambda > 0$ and $\epsilon \in \left[0, \lambda+1-\frac{1}{\alpha}\right]$, there exists a randomized $\left(\alpha(\lambda+1-\epsilon), \alpha\left(1+\frac{1}{\lambda}\right)\right)$-approximation for (k,W)-Π-Del with running time $\mathcal{O}^*\left(q^{-\frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}}\right)$.*

The key insight is that a sampling step allows us to *commit* to $z = \frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}$ vertices of an optimal solution before invoking the polynomial-time approximation. Each sampling call succeeds with probability q, so the probability that all z commitments are correct is q^z, directly giving the running time. Once S_1 of size z is fixed, Theorem 3 is applied to the reduced instance with parameters $k-z$ and $W - w(S_1)$, returning S_2. The size bound $|S| \leq \alpha(\lambda+1-\epsilon)k$ follows from $|S_2| \leq \alpha(\lambda+1)(k-z)$ and the choice of z; the weight bound is inherited unchanged from Theorem 3 since $w(S_2) \leq \alpha(1+\frac{1}{\lambda})(W - w(S_1)) \leq \alpha(1+\frac{1}{\lambda})W$.

This approach is inherently randomized. For problems where the obstruction structure is explicit, we can derandomize it entirely.

3.3 Deterministic Bi-criteria Approximation for Bounded Obstructions

For problems with bounded obstruction size, the sampling step can be replaced by deterministic branching, yielding the same guarantee without randomness.

▶ **Theorem 5** (†). *Let Π-Del admit a polynomial-time α-approximation and have obstruction size at most c. Then for any $\lambda > 0$ and $\epsilon \in \left[0, \lambda+1-\frac{1}{\alpha}\right]$, there exists a deterministic $\left(\alpha(\lambda+1-\epsilon), \alpha\left(1+\frac{1}{\lambda}\right)\right)$-approximation for (k,W)-Π-Del with running time $\mathcal{O}^*\left(c^{\frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}}\right)$.*

When obstructions have bounded size c, the randomized sampling is replaced by a deterministic c-way branch over all vertices of an obstruction H: in at least one branch, the chosen vertex belongs to an optimal solution. Branching continues until the size parameter drops below the threshold $x = \frac{(\alpha(\lambda+1-\epsilon)-1)k}{\alpha(\lambda+1)-1}$, at which point Theorem 3 is applied. The measure $\mu = k' - x$ drops by exactly 1 per branch with at most c children, so the branching tree has at most $\mathcal{O}(c^{\mu_0})$ nodes where $\mu_0 = \frac{\alpha\epsilon k}{\alpha(\lambda+1)-1}$ is the initial measure. The approximation bounds follow by the same $S = S_1 \uplus S_2$ argument as Theorem 4, with $|S_1| = k - k'$ and correctness guaranteed by the exhaustive branching rather than probabilistic sampling.

4 Bi-criteria Approximations for d-Hitting Set

We present four bi-criteria approximation algorithms for (k, W)-dHS, building progressively stronger results. The first two results improve only the size factor below d; the latter two push *both* factors simultaneously below d, a qualitatively stronger guarantee that requires a fundamentally different algorithmic technique. Full proofs and pseudocode can be found in the full version of this paper.

4.1 Polynomial-time (d, d)-Approximation

The following result serves as the base approximation invoked by all subsequent algorithms when the branching terminates.

▶ **Theorem 6** (†). *There exists a polynomial-time (d, d)-approximation algorithm for (k, W)-dHS.*

Solve the natural LP relaxation of (k, W)-dHS in polynomial time [17] and round: include e_i in S if and only if $x_i^* \geq \frac{1}{d}$. Since every set H satisfies $\sum_{e_i \in H} x_i^* \geq 1$ and $|H| \leq d$, at least one element of H has $x_i^* \geq \frac{1}{d}$, so S is a valid hitting set. The bounds $|S| \leq dk$ and $w(S) \leq dW$ follow by scaling: $1 \leq d \cdot x_i^*$ for each $e_i \in S$, and the LP budget constraints give the rest. Under the Unique Games Conjecture [18], approximating d-Hitting Set within $d - \epsilon$ is NP-hard for any $\epsilon > 0$ in polynomial time, making the factor d essentially tight and our subsequent FPT improvements particularly meaningful.

4.2 $((1 - \epsilon)d,\ d)$-Approximation

The LP rounding achieves factor d on both size and weight. We now improve the size factor below d while keeping the weight factor at d, at the cost of FPT running time.

▶ **Theorem 7** (†). *For any $\epsilon \in \left[0, \frac{d-1}{d}\right]$, there exists a $((1 - \epsilon)d,\ d)$-approximation for (k, W)-dHS with running time $\mathcal{O}^*\left(d^{\frac{\epsilon d}{d-1} \cdot k}\right)$.*

Branch on each element of an unhit set $H \in \mathcal{H}$ until the size budget drops below $x = \frac{(1-\epsilon)d-1}{d-1}k$, then invoke Theorem 6. Writing $S = S_1 \uplus S_2$ with $|S_1| = k - k'$ branching vertices and $|S_2| \leq dk'$ leaf solution vertices gives $|S| \leq k + (d - 1)k' \leq (1 - \epsilon)dk$ from the choice of x. The weight bound $w(S) \leq dW$ follows since $w(S_1) + w(S_2) \leq (W - W') + dW' = W + (d - 1)W' \leq dW$. The measure $\mu = k' - x$ drops by 1 per branch with d children, giving branching number d and running time $\mathcal{O}^*(d^{\mu_0})$ where $\mu_0 = \frac{\epsilon d}{d-1}k$.

These first two results improve size and weight *separately* – the first keeps weight at d while the second improves size below d, but neither improves both simultaneously. Achieving (a, b) with both $a < d$ and $b < d$ requires a fundamentally different branching strategy.

4.3 $\left(\frac{d+1}{2}, \frac{d+1}{2}\right)$-Approximation

The key new idea is to branch not just on one element at a time, but to simultaneously hit *two* sets in a single branching step. This is made possible by two reduction rules applied exhaustively at every node: (RR1) if a singleton $\{e\} \in \mathcal{H}$ exists, include e; (RR2) if a maximum-weight element u is dominated by some e (meaning $e \in H$ for every $H \in A_u$), remove u. After exhaustive application, a key structural property holds as an immediate consequence of RR2: for every $e \in \mathcal{U} \setminus \{u\}$, there exists $H' \in A_u$ with $e \notin H'$ (otherwise e would dominate u and RR2 would still apply). This guarantees that the pair-branches defined below are always well-defined – specifically, that $e \neq f$ in every pair-branch, so the size budget always decreases by exactly 2.

▶ **Theorem 8** (†). *There exists a* $\left(\frac{d+1}{2}, \frac{d+1}{2}\right)$*-approximation for* (k, W)*-dHS with running time* $\mathcal{O}^*\left((1+(d-1)^2)^{\frac{k}{2}}\right)$.

Proof sketch. After applying the reduction rules, branch on a maximum-weight element u via two types of steps, both incrementing a counter $\mu = \alpha + \beta$: **Branch-1** includes u (k' decreases by 1, α increases by 1); **pair-branches** include one element $e \in H \setminus \{u\}$ and one element $f \in H' \setminus \{u\}$ for some $H' \in A_u$ with $e \notin H'$ (k' decreases by 2, β increases by 1). There are at most $1 + (d-1)^2$ branches at each step. When $\mu = \frac{k}{2}$, invoke Theorem 6.

Running time. Both branch types increment μ by exactly 1, giving branching vector $\underbrace{(1, \ldots, 1)}_{1+(d-1)^2}$ and running time $\mathcal{O}^*((1 + (d-1)^2)^{k/2})$.

Approximation guarantee. Write $S = S_1 \uplus S_2$ at any leaf. A key claim: $k' \leq \alpha$. Since $\alpha + \beta = \frac{k}{2}$ and each pair-branch decreases k' by 2, we get $k' \leq k - \alpha - 2\beta = \frac{k}{2} - \beta = \alpha$.

Type-I leaf ($W' \leq \frac{W}{2}$): Using $|S_2| \leq dk'$ and $w(S_2) \leq dW'$ from Theorem 6, $|S| \leq k + (d-1)k' \leq k + (d-1)\frac{k}{2} = \frac{d+1}{2}k$ and $w(S) \leq W + (d-1)W' \leq \frac{d+1}{2}W$.

Type-II leaf ($W' > \frac{W}{2}$): Since $w(S_1) = W - W' < \frac{W}{2}$ and Branch-1 always selects a maximum-weight element, every remaining element satisfies $w(v) < \frac{W}{2\alpha}$. Hence $w(S_2) < dk' \cdot \frac{W}{2\alpha} \leq d\alpha \cdot \frac{W}{2\alpha} = \frac{dW}{2}$, giving $w(S) < \frac{W}{2} + \frac{dW}{2} = \frac{d+1}{2}W$. The size bound is identical to Type-I. ◀

4.4 $(d - \gamma, d - \gamma)$-Approximation

The $\left(\frac{d+1}{2}, \frac{d+1}{2}\right)$-approximation uses a single stopping depth $\frac{k}{2}$, locking the approximation factor at $\frac{d+1}{2}$ with no flexibility. By introducing separate size and weight thresholds x and Y parametrized by γ, branching can stop earlier on some branches and later on others depending on which budget is exhausted first, yielding a continuous family of $(d-\gamma, d-\gamma)$-approximations.

▶ **Theorem 9** (†). *For any* $\gamma \in \left(0, \frac{d-1}{2}\right)$*, there exists a* $(d - \gamma, d - \gamma)$*-approximation for* (k, W)*-dHS with running time* $\mathcal{O}^*(M_{\gamma,d}^k)$*, where:*

$$M_{\gamma,d} = \max_{t \in [t_{\text{cross}}, t_{\max}]} f_{\gamma,d}(t), \quad f_{\gamma,d}(t) = \frac{(\delta_0 + t)^{\delta_0 + t} \cdot (d-1)^{2(1+r)t}}{(\delta_0 - rt)^{\delta_0 - rt} \cdot ((1+r)t)^{(1+r)t}}, \tag{1}$$

with $\delta_0 = \frac{\gamma}{d-1}$, $r = \frac{2\gamma}{d-2\gamma-1}$, $t_{\text{cross}} = \frac{\gamma(d-2\gamma-1)}{1+2\gamma(d-1)}$, $t_{\max} = \frac{d-2\gamma-1}{2(d-1)}$.

Proof sketch. Use the same $(1 + (d-1)^2)$-way branching with two global thresholds $x = \frac{d-\gamma-1}{d-1}k$ and $Y = \frac{d-\gamma-1}{d-1}W$. Nodes are classified into four types based on the current budgets k' and W': **Type-I** ($k' \leq x$, $W' \leq Y$) and **Type-II** ($k' \leq x$, $W' > Y$, no element of weight $\geq Y/k'$) are stopping nodes where Theorem 6 is invoked; **Type-IV** ($k' > x$) and **Type-III** ($k' \leq x$, $W' > Y$, some element of weight $\geq Y/k'$ exists) continue branching.

Approximation guarantee. At Type-I leaves, the same calculation as Theorem 8 gives $|S| \leq (d-\gamma)k$ and $w(S) \leq (d-\gamma)W$ directly from $k' \leq x$ and $W' \leq Y$. At Type-II leaves, $w(S_1) < \frac{\gamma}{d-1}W$ (since $W' > Y$) and $w(S_2) < dY = \frac{d(d-\gamma-1)}{d-1}W$, giving $w(S) < (d-\gamma)W$ by a direct algebraic computation.

Running time. Type-IV nodes appear only in the first $\delta_0 k$ layers (†), contributing $\mathcal{O}^*((1+(d-1)^2)^{\delta_0 k})$. Type-III nodes appear in layers $\delta_0 k + i$ for $0 \le i \le i_{\max}$ (†). The key structural result (†) is that at layer $\delta_0 k + i$, any Type-III node has at most $A = \lfloor \delta_0 k - ri \rfloor$ Branch-1 steps, so the node count at that layer is bounded by $\sum_{\alpha=0}^{A} \binom{\ell}{\alpha}(d-1)^{2(\ell-\alpha)}$. For small $i \le i_{\text{cross}}$ this partial sum equals the full binomial $(1+(d-1)^2)^\ell$; for large i it is dominated by the term at $\alpha = A$, and a Stirling approximation gives $\mathcal{O}^*(f_{\gamma,d}(i/k)^k)$. Taking the maximum over all layers yields $\mathcal{O}^*(M_{\gamma,d}^k)$. ◀

Numerical values of $M_{\gamma,d}$ and their corresponding plots are given in Table 2 and Figure 1 respectively.

Table 2 Selected values of $M_{\gamma,d}$. Boundary rows ($\gamma = \frac{d-1}{2}$) correspond to Theorem 8. Full table in the full version.

d	γ	Approx. factor	$M_{\gamma,d}$
	0.25	$(1.75, 1.75)$	1.2720
2	0.40	$(1.60, 1.60)$	1.3626
	0.50	$(1.50, 1.50)$	$\sqrt{2}$
	0.50	$(2.50, 2.50)$	2.0571
3	0.75	$(2.25, 2.25)$	2.1440
	1.00	$(2.00, 2.00)$	$\sqrt{5}$
	0.50	$(3.50, 3.50)$	3.0030
4	1.00	$(3.00, 3.00)$	3.0540
	1.50	$(2.50, 2.50)$	$\sqrt{10}$

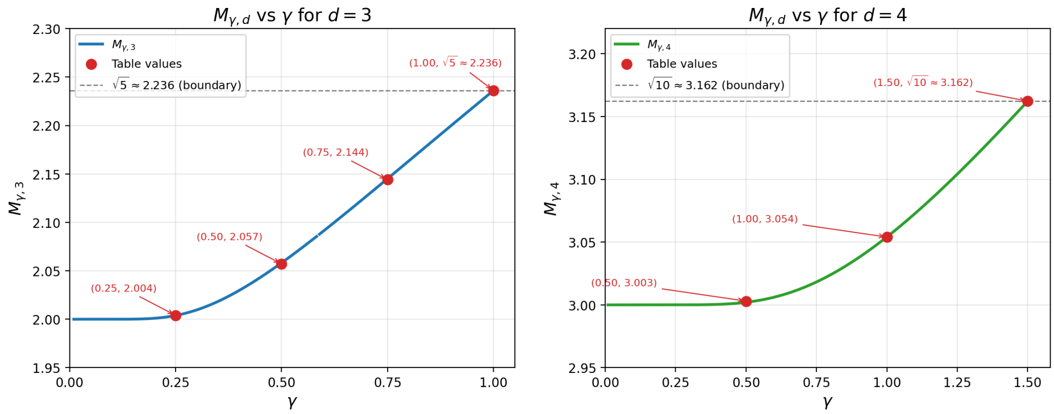

Figure 1 $M_{\gamma,d}$ vs γ for $d = 3$ (left) and $d = 4$ (right). Dashed lines mark boundary values $\sqrt{5}$ and $\sqrt{10}$ at $\gamma = \frac{d-1}{2}$.

5 Applications

Our results apply to any vertex deletion problem admitting a polynomial-time weighted α-approximation. Problems naturally fall into two categories: those with bounded obstruction size d, where the d-Hitting Set results apply directly and – when $\alpha < d$ – the general results of Section 3 provide additional trade-offs; and those with unbounded obstructions, where a sampling step is needed. Full application tables appear in the full version.

5.1 Problems with Bounded Obstructions

Cluster Vertex Deletion (CVD). The unique obstruction is P_3, giving $d = 3$. A polynomial-time 2-approximation exists for weighted CVD [2], so $\alpha = 2 < d$ and $c = 3$. All four d-Hitting Set theorems apply, as does Theorem 5. Selected results are in Table 3 (left).

FVS in Tournaments (FVST). The obstructions are directed triangles, giving $d = 3$. A randomized polynomial-time 2-approximation exists [22], so $\alpha = 2$ and $c = 3$. The approximation factors and running times are identical to CVD; the only difference is that all results are randomized, inheriting the randomness of the underlying approximation.

Split Vertex Deletion (SVD). The obstructions are C_4, C_5, $2K_2$, giving $d = 5$. A polynomial-time $(2 + \sigma)$-approximation exists [8] for any $\sigma > 0$, so $\alpha = 2 + \sigma$ and $c = 5$. All four d-Hitting Set results apply, and Theorem 3 gives additional polynomial-time results with weight factor below 5. Selected results are in Table 4 (right).

Table 3 (k, W)-CVD ($d = 3$, $\alpha = 2$, $c = 3$).

(a, b)	Param.	Time	Thm.
$(3, 3)$	–	poly	6
$(2, 3)$	$\epsilon = 1/3$	$\mathcal{O}^*(1.732^k)$	7
$(2, 2)$	–	$\mathcal{O}^*(2.236^k)$	8
$(2, 4)$	$\lambda = 1, \epsilon = 1$	$\mathcal{O}^*(2.080^k)$	5

Table 4 (k, W)-SVD ($d = 5$, $\alpha \approx 2$, $c = 5$).

(a, b)	Param.	Time	Thm.
$(5, 5)$	–	poly	6
$(4, 4)$	$\lambda = 1$	poly	3
$(3, 3)$	–	$\mathcal{O}^*(4.123^k)$	8
$(2, 4)$	$\lambda = 1, \epsilon = 1$	$\mathcal{O}^*(2.924^k)$	5

d-Path Vertex Cover (d-PVC). The obstructions are paths P_d, so the maximum obstruction size is exactly d, and all four d-Hitting Set results apply for any $d \geq 2$. For $d = 3$, a 2-approximation exists [30] ($\alpha = 2$, $c = 3$), giving results identical to CVD. For $d = 4$, a 3-approximation exists [5] ($\alpha = 3$, $c = 4$), and Theorem 5 applies with modest additional gain since α is close to d. For $d \geq 5$, only the d-Hitting Set results apply.

5.2 Problems with Unbounded Obstructions

Feedback Vertex Set (FVS). The obstructions are cycles of all lengths, so the d-Hitting Set results do not apply. We establish a sampling step via a careful edge-counting argument after two exhaustive reduction rules (deleting degree-≤ 1 vertices and contracting long degree-2 paths).

▶ **Lemma 10.** *(k, W)-FVS admits a sampling step with probability at least $\frac{1}{8}$.*

Proof. We first apply two reduction rules exhaustively to the input instance (G, w, k, W).

▶ **Reduction Rule 1.** *For an instance (G, w, k, W) of (k, W)-FVS, if there is a vertex v in G of degree at most one, then delete the vertex. The new instance will be $(G - \{v\}, w, k, W)$.*

This is correct since any vertex of degree ≤ 1 participates in no cycle, so no minimal FVS contains it.

▶ **Reduction Rule 2.** *For an instance (G, w, k, W) of (k, W)-FVS, if there is a maximal degree-2 path $P = (u, x_1, x_2, \ldots, x_l, v)$ of length $i{+}2$ with $i \geq 2$, then replace path P by another path $P' = (u, y, v)$ where $y \in X = \{x_1, x_2, \ldots, x_l\}$ such that $w(y) = \min\{w(x_i) \mid 1 \leq i \leq l\}$.*

This is correct: let $X = \{x_1, \ldots, x_\ell\}$. For the forward direction, given a feasible FVS S for (G, w, k, W), if $S \cap X = \emptyset$ or $y \in S$ then S works unchanged for the reduced instance. Otherwise let $S' = (S \setminus X) \cup \{y\}$; since all vertices of X lie on the same cycles, S' is a valid FVS with $|S'| \le |S| \le k$ and $w(S') \le w(S) \le W$. The reverse direction holds since subdividing an edge creates no new cycle.

Sampling step $\mathcal{R}$. After exhaustive reduction, pick a uniformly random edge of the resulting graph $G' = (V', E')$ and return one of its two endpoints uniformly at random.

Analysis. We give the proof of the Lemma 10 by proving that for any fvs S of size at most k and weight at most W, the edge picked by $\mathcal{R}$ has at least one end vertex in S with probability at least $\frac{1}{4}$.

We prove the claim by showing that if none of the reduction rules is applicable, then for any feedback vertex (FVS) of size at most k and weight at most W in the graph, the edge selected by $\mathcal{R}$ has at least one endpoint in the FVS with probability at least $\frac{1}{4}$.

Let $G' = (V', E')$ be the graph obtained after applying the reduction rules, and let F be a feedback vertex set of G' with $|F| \le k$ and weight at most W. Define

$$E_F := E'(F) \cup E'(F, V' \setminus F), \qquad E_T := E'(V' \setminus F),$$

where $E'(A)$ denotes the set of edges of G' with both endpoints in A, and $E'(A, B)$ denotes the set of edges with one endpoint in A and the other in B. Since F is a feedback vertex set, the induced subgraph $T := G'[V' \setminus F]$ is a forest. We will establish that $|E_F| \ge \frac{1}{3}|E_T|$, which implies $\frac{|E_F|}{|E'|} \ \ge \ \frac{1}{4}$.

We partition the leaf vertices of T into the following sets:

$$L_2 = \{v \in T : \deg_T(v) = 1,\ \deg_{G'}(v) = 2\}, \quad L_{\ge 3} = \{v \in T : \deg_T(v) = 1,\ \deg_{G'}(v) \ge 3\}.$$

Similarly, partition the internal vertices of T as:

$$I_2 = \{v \in T : \deg_T(v) = 2,\ \deg_{G'}(v) = 2\}, \qquad I_2' = \{v \in T : \deg_T(v) = 2,\ \deg_{G'}(v) \ge 3\},$$
$$I_{\ge 3} = \{v \in T : \deg_T(v) \ge 3\}.$$

Let $x = |L_2|$, $y = |L_{\ge 3}|$, $z = |I_2'|$, $p = |I_2|$, and $r = |I_{\ge 3}|$.

Each $v \in L_2$ contributes at least one edge to E_F, each $v \in L_{\ge 3}$ contributes at least two, and each $v \in I_2'$ contributes at least one. Thus

$$|E_F| \ \ge \ x + 2y + z. \tag{2}$$

Since none of the reduction rules are applicable, in the forest T, each vertex in I_2 will have a unique child in $L_{\ge 3} \cup I_{\ge 3} \cup I_2'$, so $p \le y + r + z$. Moreover, in any forest, we have $r \le x + y$. Therefore

$$p \ \le \ x + 2y + z.$$

Hence, the total number of vertices in T satisfies

$$n_T = x + y + p + z + r \ \le \ x + y + (x + 2y + z) + z + (x + y) = 3x + 4y + 2z.$$

Since T is a forest, $|E_T| \le n_T$, so

$$|E_T| \ \le \ 3x + 4y + 2z. \tag{3}$$

Combining (2) and (3), we obtain

$$|E_F| \;\geq\; x + 2y + z \;\geq\; \tfrac{1}{3}(3x + 6y + 3z) \;\geq\; \tfrac{1}{3}(3x + 4y + 2z) \;\geq\; \tfrac{1}{3}|E_T|.$$

Therefore $|E_F|/|E'| \geq \frac{1}{4}$, i.e., a random edge of G' touches F with probability at least $\frac{1}{4}$. This completes the proof. ◀

Combined with the 2-approximation of [3] ($\alpha = 2$, $q = \frac{1}{8}$), Theorems 3 and 4 give the results in Table 5 (left).

Pathwidth-One Vertex Deletion (POVD). The obstructions are cycles and the graph T_2 – a tree with three paths of length two sharing a common internal vertex, giving $|V(T_2)| = 7$. We first establish a polynomial-time 3-approximation.

▶ **Theorem 11.** *There exists a polynomial-time* 3*-approximation for weighted* POVD *on general graphs.*

Proof. Compute a 2-approximate weighted FVS S_1 via [3], giving $w(S_1) \leq 2w(S^*)$. Since graphs of pathwidth at most one are acyclic [29], S^* is also a valid FVS. Solve weighted POVD exactly on the forest $G - S_1$ (Theorem 12), giving S_2 with $w(S_2) \leq w(S^* \cap V(G - S_1)) \leq w(S^*)$. Return $S_1 \cup S_2$; then $w(S_1 \cup S_2) \leq 3w(S^*)$. ◀

The exact algorithm for weighted POVD on forests (used in Phase 2 above) is a tree DP assigning each vertex a state in {deleted, leaf, spine}, tracking parent-spine status via a binary flag, and enforcing local caterpillar constraints via a constant-capacity knapsack at each node.

▶ **Theorem 12** (†)**.** *Weighted* POVD *on forests is solvable in $O(n)$ time via tree dynamic programming.*

We also establish a sampling step for (k, W)-POVD.

▶ **Lemma 13.** (k, W)-POVD *admits a sampling step with probability at least $\frac{1}{7}$.*

Proof. If G contains T_2, K_3, or C_4, find one and return a uniformly random vertex; since $|V(H)| \leq 7$, success probability $\geq \frac{1}{7}$. Otherwise, since G contains no T_2, K_3, or C_4, every component of G is a tree or a unicyclic graph [26]; return the minimum-weight vertex of any cycle (success probability $1 \geq \frac{1}{7}$). ◀

With $\alpha = 3$ and $q = \frac{1}{7}$, Theorems 3 and 4 give the results in Table 6 (right).

Table 5 (k, W)-FVS ($\alpha = 2$, $q = \frac{1}{8}$).
* randomized.

(a, b)	Param.	Time	Thm.
$(4, 4)$	$\lambda = 1$	poly	3
$(3.5, 4)$	$\lambda = 1, \epsilon = 0.25$	$\mathcal{O}^*(1.414^k)$	4*
$(2.5, 4)$	$\lambda = 1, \epsilon = 0.75$	$\mathcal{O}^*(2.828^k)$	4*
$(9.5, 2.5)$	$\lambda = 4, \epsilon = 0.25$	$\mathcal{O}^*(1.122^k)$	4*

Table 6 (k, W)-POVD ($\alpha = 3$, $q = \frac{1}{7}$).
* randomized.

(a, b)	Param.	Time	Thm.
$(6, 6)$	$\lambda = 1$	poly	3
$(4.5, 6)$	$\lambda = 1, \epsilon = 0.5$	$\mathcal{O}^*(1.793^k)$	4*
$(3, 6)$	$\lambda = 1, \epsilon = 1$	$\mathcal{O}^*(3.214^k)$	4*
$(6, 4.5)$	$\lambda = 2, \epsilon = 1$	$\mathcal{O}^*(2.074^k)$	4*

6 Conclusion

We studied bi-criteria approximation algorithms for (k, W)-Π-DEL and (k, W)-d-HITTING SET, simultaneously approximating both solution size and weight in the FPT setting. Our results range from polynomial-time LP-rounding to FPT branching algorithms, and apply broadly to any vertex deletion problem admitting a weighted approximation. For problems with bounded obstruction size, we additionally obtain algorithms that push *both* approximation factors simultaneously below d, a guarantee that polynomial-time algorithms cannot achieve under the Unique Games Conjecture. We leave the following directions open.

1. **Tight running time for $d \geq 3$.** The bound $O^*(M^k_{\gamma,d})$ is tight for $d = 2$ but potentially loose for $d \geq 3$. To achieve the $(d - \gamma, d - \gamma)$ guarantee on both size and weight, the algorithm must continue branching on Type-III instances even after the size budget drops below $x = \frac{d-\gamma-1}{d-1}k$, all the way up to layer $\frac{k}{2}$. At each such step there are $(d - 1)^2$ pair-branch children, each of which can independently remain Type-III and continue branching, producing $(d - 1)^{2(\ell-\alpha)}$ distinct paths at layer ℓ with α Branch-1 steps. This is not an artifact of analysis – it is a genuine consequence of the branching structure. For $d = 2$, $(d - 1)^2 = 1$ so there is only one pair-branch child at each step, and the Type-III node count at each layer is a single binomial coefficient with no exponential factor. For $d \geq 3$, even with zero Branch-1 steps the count is $(d - 1)^{2\ell}$, which is exponentially large. A technique that avoids or reduces the $(1 + (d - 1)^2)$-way branching is needed for a tight analysis.
2. **Hardness of bi-criteria approximation.** Establishing lower bounds on the achievable factor pair (a, b) in polynomial time or FPT time would complement our algorithms and clarify the limits of the bi-criteria framework.
3. **Sampling steps and direct branching.** For FVS and POVD we relied on sampling steps with probabilities $\frac{1}{8}$ and $\frac{1}{7}$. Improving these probabilities, or replacing sampling entirely with direct branching techniques, would strengthen the corresponding bi-criteria results.
4. **Weighted POVD approximation ratio.** Our polynomial-time 3-approximation for weighted POVD matches the best known ratio for the unweighted version [16]. Whether a ratio below 3 is achievable remains open.

References

1 Akanksha Agrawal, Sudeshna Kolay, Daniel Lokshtanov, and Saket Saurabh. A faster FPT algorithm and a smaller kernel for block graph vertex deletion. In Evangelos Kranakis, Gonzalo Navarro, and Edgar Chávez, editors, *LATIN 2016: Theoretical Informatics - 12th Latin American Symposium, Ensenada, Mexico, April 11-15, 2016, Proceedings*, volume 9644 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 2016. doi:10.1007/978-3-662-49529-2_1.

2 Manuel Aprile, Matthew Drescher, Samuel Fiorini, and Tony Huynh. A tight approximation algorithm for the cluster vertex deletion problem. *Math. Program.*, 197(2):1069–1091, 2023. doi:10.1007/S10107-021-01744-W.

3 Vineet Bafna, Piotr Berman, and Toshihiro Fujito. A 2-approximation algorithm for the undirected feedback vertex set problem. *SIAM J. Discret. Math.*, 12(3):289–297, 1999. doi:10.1137/S0895480196305124.

4 Ljiljana Brankovic and Henning Fernau. A novel parameterised approximation algorithm for minimum vertex cover. *Theor. Comput. Sci.*, 511:85–108, 2013. doi:10.1016/j.tcs.2012.12.003.

5 Eglantine Camby, Jean Cardinal, Mathieu Chapelle, Samuel Fiorini, and Gwenaël Joret. A primal-dual 3-approximation algorithm for hitting 4-vertex paths. In *9th International Colloquium on Graph Theory and Combinatorics (ICGT 2014)*, page 61, 2014.

6 Jianer Chen, Fedor V. Fomin, Yang Liu, Songjian Lu, and Yngve Villanger. Improved algorithms for the feedback vertex set problems. In Frank K. H. A. Dehne, Jörg-Rüdiger Sack, and Norbert Zeh, editors, *Algorithms and Data Structures, 10th International Workshop, WADS 2007, Halifax, Canada, August 15-17, 2007, Proceedings*, volume 4619 of *Lecture Notes in Computer Science*, pages 422–433. Springer, 2007. doi:10.1007/978-3-540-73951-7_37.

7 Marek Cygan, Fedor V. Fomin, Lukasz Kowalik, Daniel Lokshtanov, Dániel Marx, Marcin Pilipczuk, Michal Pilipczuk, and Saket Saurabh. *Parameterized Algorithms*. Springer, 2015. doi:10.1007/978-3-319-21275-3.

8 Matthew Drescher, Samuel Fiorini, and Tony Huynh. A simple $(2+\epsilon)$-approximation algorithm for split vertex deletion. *Eur. J. Comb.*, 121:103844, 2024. doi:10.1016/J.EJC.2023.103844.

9 Baris Can Esmer and Ariel Kulik. Sampling with a black box: Faster parameterized approximation algorithms for vertex deletion problems. In Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis, editors, *52nd International Colloquium on Automata, Languages, and Programming, ICALP 2025, Aarhus, Denmark, July 8-11, 2025*, LIPIcs, pages 39:1–39:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.ICALP.2025.39.

10 Andreas Emil Feldmann, Karthik C. S., Euiwoong Lee, and Pasin Manurangsi. A survey on approximation in parameterized complexity: Hardness and algorithms. *Algorithms*, 13(6):146, 2020. doi:10.3390/a13060146.

11 Michael R. Fellows, Ariel Kulik, Frances A. Rosamond, and Hadas Shachnai. Parameterized approximation via fidelity preserving transformations. *J. Comput. Syst. Sci.*, 93:30–40, 2018. doi:10.1016/j.jcss.2017.11.001.

12 Esther Galby, Dániel Marx, Philipp Schepper, Roohani Sharma, and Prafullkumar Tale. Parameterized complexity of weighted multicut in trees. In Michael A. Bekos and Michael Kaufmann, editors, *Graph-Theoretic Concepts in Computer Science - 48th International Workshop, WG 2022, Tübingen, Germany, June 22-24, 2022, Revised Selected Papers*, volume 13453 of *Lecture Notes in Computer Science*, pages 257–270. Springer, 2022. doi:10.1007/978-3-031-15914-5_19.

13 David G. Harris and N. S. Narayanaswamy. A faster algorithm for vertex cover parameterized by solution size. In Olaf Beyersdorff, Mamadou Moustapha Kanté, Orna Kupferman, and Daniel Lokshtanov, editors, *41st International Symposium on Theoretical Aspects of Computer Science, STACS 2024, March 12-14, 2024, Clermont-Ferrand, France*, volume 289 of *LIPIcs*, pages 40:1–40:18. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2024. doi:10.4230/LIPIcs.STACS.2024.40.

14 Yoichi Iwata and Yusuke Kobayashi. Improved analysis of highest-degree branching for feedback vertex set. *Algorithmica*, 83(8):2503–2520, 2021. doi:10.1007/S00453-021-00815-W.

15 Satyabrata Jana, Daniel Lokshtanov, Soumen Mandal, Ashutosh Rai, and Saket Saurabh. Parameterized approximation scheme for feedback vertex set. In *48th International Symposium on Mathematical Foundations of Computer Science (MFCS 2023)*, volume 272 of *LIPIcs*, pages 56:1–56:15, 2023. doi:10.4230/LIPIcs.MFCS.2023.56.

16 Satyabrata Jana, Soumen Mandal, Ashutosh Rai, and Saket Saurabh. Improved approximation for pathwidth one vertex deletion and parameterized complexity of its variants. In C. Aiswarya, Ruta Mehta, and Subhajit Roy, editors, *45th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2025, BITS Pilani, K K Birla Goa Campus, India, December 17-19, 2025*, LIPIcs, pages 39:1–39:20. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, 2025. doi:10.4230/LIPIcs.FSTTCS.2025.39.

17 Narendra Karmarkar. A new polynomial-time algorithm for linear programming. *Comb.*, 4(4):373–396, 1984. doi:10.1007/BF02579150.

18 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within 2-epsilon. *J. Comput. Syst. Sci.*, 74(3):335–349, 2008. doi:10.1016/J.JCSS.2007.06.019.

19 Eun Jung Kim, Tomás Masarík, Marcin Pilipczuk, Roohani Sharma, and Magnus Wahlström. On weighted graph separation problems and flow augmentation. *SIAM J. Discret. Math.*, 38(1):170–189, 2024. doi:10.1137/22M153118X.

20 Ariel Kulik and Hadas Shachnai. Analysis of two-variable recurrence relations with application to parameterized approximations. In *61st IEEE Annual Symposium on Foundations of Computer Science (FOCS 2020)*, pages 762–773. IEEE, 2020. doi:10.1109/FOCS46700.2020.00076.

21 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *J. Comput. Syst. Sci.*, 20(2):219–230, 1980. doi:10.1016/0022-0000(80)90060-4.

22 Daniel Lokshtanov, Pranabendu Misra, Joydeep Mukherjee, Fahad Panolan, Geevarghese Philip, and Saket Saurabh. 2-approximating feedback vertex set in tournaments. *ACM Trans. Algorithms*, 17(2):11:1–11:14, 2021. doi:10.1145/3446969.

23 Soumen Mandal, Pranabendu Misra, Ashutosh Rai, and Saket Saurabh. Parameterized approximation algorithms for weighted vertex cover. *Theor. Comput. Sci.*, 1021:114870, 2024. doi:10.1016/J.TCS.2024.114870.

24 George L. Nemhauser and Leslie E. Trotter Jr. Vertex packings: Structural properties and algorithms. *Math. Program.*, 8(1):232–248, 1975. doi:10.1007/BF01580444.

25 Rolf Niedermeier and Peter Rossmanith. On efficient fixed-parameter algorithms for weighted vertex cover. *J. Algorithms*, 47(2):63–77, 2003. doi:10.1016/S0196-6778(03)00005-1.

26 Geevarghese Philip, Venkatesh Raman, and Yngve Villanger. A quartic kernel for pathwidth-one vertex deletion. In Dimitrios M. Thilikos, editor, *Graph Theoretic Concepts in Computer Science - 36th International Workshop, WG 2010, Zarós, Crete, Greece, June 28-30, 2010 Revised Papers*, volume 6410 of *Lecture Notes in Computer Science*, pages 196–207, 2010. doi:10.1007/978-3-642-16926-7_19.

27 Hadas Shachnai and Meirav Zehavi. A multivariate framework for weighted FPT algorithms. *J. Comput. Syst. Sci.*, 89:157–189, 2017. doi:10.1016/j.jcss.2017.05.003.

28 Dekel Tsur. Faster parameterized algorithm for cluster vertex deletion. *Theory Comput. Syst.*, 65(2):323–343, 2021. doi:10.1007/S00224-020-10005-W.

29 Dekel Tsur. Faster algorithm for pathwidth one vertex deletion. *Theor. Comput. Sci.*, 921:63–74, 2022. doi:10.1016/J.TCS.2022.04.001.

30 Jianhua Tu and Wenli Zhou. A factor 2 approximation algorithm for the vertex cover p_3 problem. *Inf. Process. Lett.*, 111(14):683–686, 2011. doi:10.1016/J.IPL.2011.04.009.