

On Deleting Vertices to Reduce Density in Graphs and Supermodular Functions

Karthekeyan Chandrasekaran  

Grainger College of Engineering, University of Illinois, Urbana-Champaign, IL, USA

Chandra Chekuri 

Grainger College of Engineering, University of Illinois, Urbana-Champaign, IL, USA

Shubhang Kulkarni  

Grainger College of Engineering, University of Illinois, Urbana-Champaign, IL, USA

Abstract

We consider deletion problems in graphs and supermodular functions where the goal is to reduce density. In Graph Density Deletion (GRAPHDD), we are given a graph $G = (V, E)$ with non-negative vertex costs and a non-negative parameter $\rho \geq 0$ and the goal is to remove a minimum cost subset S of vertices such that the densest subgraph in $G - S$ has density at most ρ . This problem has an underlying matroidal structure and generalizes several classical problems such as vertex cover, feedback vertex set, and pseudoforest deletion set for appropriately chosen $\rho \leq 1$ and all of these classical problems admit a 2-approximation. In sharp contrast, we prove that for every fixed integer $\rho > 1$, GRAPHDD is hard to approximate to within a logarithmic factor via a reduction from SETCOVER, thus showing a phase transition phenomenon. Next, we investigate a generalization of GRAPHDD to monotone supermodular functions, termed Supermodular Density Deletion (SUPMODDD). In SUPMODDD, we are given a monotone supermodular function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ via an evaluation oracle with element costs and a non-negative integer $\rho \geq 0$ and the goal is to remove a minimum cost subset $S \subseteq V$ such that the densest subset according to f in $V - S$ has density at most ρ . We show that SUPMODDD is approximation equivalent to the well-known SUBMODULAR COVER problem; this implies a tight logarithmic approximation and hardness for SUPMODDD; it also implies a logarithmic approximation for GRAPHDD, thus matching our inapproximability bound. Motivated by these hardness results, we design bicriteria approximation algorithms for both GRAPHDD and SUPMODDD.

2012 ACM Subject Classification Theory of computation \rightarrow Graph algorithms analysis; Theory of computation \rightarrow Approximation algorithms analysis

Keywords and phrases Combinatorial Optimization, Approximation Algorithms, Randomized Algorithms, Hardness of Approximation, Densest Subgraph, Supermodular Functions, Submodular Set Cover

Digital Object Identifier 10.4230/LIPIcs.ICALP.2025.43

Category Track A: Algorithms, Complexity and Games

Related Version Full version of this work is available at: <https://arxiv.org/abs/2503.08828> [6]

Funding This work was supported in part by NSF grant CCF-2402667.

Acknowledgements Shubhang Kulkarni thanks Kishen Gowda for engaging in preliminary discussions about approximations for feedback vertex set and pseudoforest deletion set. We thank anonymous reviewers for pointing out an error, and for suggesting improvements to a previous version of this work.

1 Introduction

The *densest subgraph* problem in graphs (DSG) is a core primitive in graph and network mining applications. In DSG, we are given a graph $G = (V, E)$ and the goal is to find $\lambda_G^* := \max_{S \subseteq V} |E(S)|/|S|$, where $E(S)$ is the set of edges with both end vertices in S .



© Karthekeyan Chandrasekaran, Chandra Chekuri, and Shubhang Kulkarni; licensed under Creative Commons License CC-BY 4.0

52nd International Colloquium on Automata, Languages, and Programming (ICALP 2025).

Editors: Keren Censor-Hillel, Fabrizio Grandoni, Joël Ouaknine, and Gabriele Puppis

Article No. 43; pp. 43:1–43:20



Leibniz International Proceedings in Informatics

Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany



DSG is not only interesting for its applications but is a fundamental problem in algorithms and combinatorial optimization with several connections to graph theory, matroids, and submodularity. Many recent works have explored various aspects of DSG and related problems from both theoretical and practical perspectives [4, 8, 9, 12, 13, 20, 23, 26, 28, 30]. A useful feature of DSG is its polynomial-time solvability. This was first seen via a reduction to network flow [19, 27] but another way to see it is by considering a more general problem, namely the *densest supermodular subset* problem (DSS): Given a supermodular function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$ via evaluation oracle, the goal is to find $\lambda_f^* := \max_{S \subseteq V} f(S)/|S|$. One can easily see that DSG is a special case of DSS by noting that for any graph G , the function $f : 2^V \rightarrow \mathbb{Z}$ defined by $f(S) = |E(S)|$ for every $S \subseteq V$ is a supermodular function. It is well-known and easy to see that DSS and DSG can be solved in polynomial-time by a simple reduction to submodular function minimization. Several other problems that are studied in graph and network mining can be seen as special cases of DSS. Recent work has demonstrated the utility of the supermodularity lens in understanding greedy heuristics and approximation algorithms for DSG and these problems [9, 20, 21, 30].

Density Deletion Problems. In this work we consider several interrelated *vertex deletion* problems that aim to *reduce* the density. We start with the graph density deletion problem.

► **Definition 1** (ρ -GRAPHDD). *For a fixed constant ρ , the ρ -graph density deletion problem, denoted ρ -GRAPHDD, is defined as follows:*

Input: Graph $G = (V, E)$ and vertex costs $c : V \rightarrow \mathbb{R}_{\geq 0}$
Goal: $\arg \min\{\sum_{u \in S} c_u : S \subseteq V \text{ and } \lambda_{G-S}^* \leq \rho\}$.

This deletion problem naturally generalizes to supermodular functions as defined below. We recall that a set function $f : 2^V \rightarrow \mathbb{R}$ is (i) submodular if $f(A) + f(B) \geq f(A \cap B) + f(A \cup B)$ for every $A, B \subseteq V$, (ii) supermodular if $-f$ is submodular, (iii) non-decreasing if $f(A) \leq f(B)$ for every $A \subseteq B \subseteq V$, and (iv) normalized if $f(\emptyset) = 0$. We observe that non-negative normalized supermodular functions are non-decreasing. For a function $f : 2^V \rightarrow \mathbb{R}$ and $S \subseteq V$, we define f_{V-S} as the function f restricted to the ground set $V - S$. The evaluation oracle for the function takes a subset $S \subseteq V$ as input and returns the function value of the set S .

► **Definition 2** (ρ -SUPMODDD). *For a fixed constant ρ , the ρ -supermodular density deletion problem, denoted ρ -SUPMODDD, is defined as follows:*

Input: Integer-valued normalized supermodular function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ via an evaluation oracle and element costs $c : V \rightarrow \mathbb{R}_{\geq 0}$
Goal: $\arg \min\{\sum_{u \in S} c_u : S \subseteq V \text{ and } \lambda_{f_{V-S}}^* \leq \rho\}$.

When the density threshold ρ is part of input, we use GRAPHDD and SUPMODDD to refer to these problems. It is easy to see that GRAPHDD (and hence SUPMODDD) is NP-Hard from a general result on vertex deletion problems [25]. Our goal is to understand the approximability of these problems.

Motivations and Connections. While the deletion problems are natural in their formulation, to the best of our knowledge, GRAPHDD has only recently been explicitly defined and explored. Bazgan, Nichterlein and Vazquez Alferrez [2] defined and studied this problem from an FPT

perspective. As pointed out in their work, given the importance of DSG and DSS in various applications to detect communities and sub-groups of interest, it is useful to consider the robustness (or sensitivity) of the densest subgraph to the removal of vertices. In this context, we mention the classical work of Cunningham on the attack problem [11] which can be seen as the problem of deleting *edges* to reduce density; this edge deletion problem can be solved in polynomial time for integer parameters ρ via matroidal and network flow techniques. In addition to their naturalness and the recent work, we are motivated to consider GRAPHDD and SUPMODDD owing to their connections to several classical vertex deletion problems as well as a matroidal structure underlying GRAPHDD that we articulate next.

We observe that 0-GRAPHDD is equivalent to the vertex cover problem: requiring density of 0 after deleting S is equivalent to S being a vertex cover of G . One can also see, in a similar fashion, that 1-GRAPHDD is equivalent to the pseudoforest deletion set problem, denoted PFDS— where the goal is to delete vertices so that every connected component in the remaining graph has at most one cycle, and $(1 - 1/|V|)$ -GRAPHDD is equivalent to the feedback vertex set problem, denoted FVS— where the goal is to delete vertices so that the remaining graph is acyclic (see [5] for formal definitions of both problems). Vertex cover, PFDS, and FVS admit 2-approximations, and moreover this bound cannot be improved under the Unique Games Conjecture (UGC) [22]. We note that while 2-approximations for vertex cover are relatively easy, 2-approximations for FVS and PFDS are non-obvious [1, 3, 10]. Until very recently there was no polynomial-time solvable linear program (LP) that yielded a 2-approximation for FVS and PFDS. In fact, the new and recent LP formulations [5] for FVS and PFDS are obtained via connections to Charikar’s LP-relaxation for DSG [7]. Fujito [18] unified the 2-approximations for vertex cover, FVS, and PFDS via primal-dual algorithms by considering a more general class of *matroidal* vertex deletion problem on graphs that is relevant to our work. This abstract problem, denoted MATROIDFVS¹, is defined below.

► **Definition 3** (MATROIDFVS). *The Matroid Feedback Vertex Set problem, denoted MATROIDFVS, is defined as follows:*

Input: Graph $G = (V, E)$, vertex costs $c : V \rightarrow \mathbb{R}_{\geq 0}$, and
 Matroid $\mathcal{M} = (E, \mathcal{I})$ with \mathcal{I} being the collection of independent sets
 (via an independence testing oracle)

Goal: $\arg \min \{ \sum_{u \in S} c_u : S \subseteq V \text{ and } E[V - S] \in \mathcal{I} \}$.

Fujito [18] obtained a 2-approximation for MATROIDFVS for the class of “uniformly sparse” matroids [24]. It is not difficult to show that vertex cover, FVS, and PFDS can be cast as special cases of MATROIDFVS where the associated matroids are “uniformly sparse”. Consequently, Fujito’s result unifies the 2-approximations for these three fundamental problems.

We now observe some non-trivial connections between ρ -GRAPHDD, MATROIDFVS and ρ -SUPMODDD. We can show that ρ -GRAPHDD is a special case of MATROIDFVS for every integer ρ : indeed, ρ -GRAPHDD corresponds to MATROIDFVS where the matroid \mathcal{M}_ρ is the ρ -fold union of the 1-cycle matroid defined on the edge set of the input graph. Although it is not obvious, we can show that MATROIDFVS is a special case of 1-SUPMODDD. We refer the

¹ We use the *feedback vertex set* terminology in our naming of the MATROIDFVS problem since the goal is to pick a min-cost subset of vertices to cover all *circuits* of the matroid defined on the edges of a graph. This generalizes FVS which is MATROIDFVS where the matroid of interest is the graphic matroid on the input graph.

reader to Appendix A of the full version [6] of this work for details regarding these connections, and the problems in the right column in Figure 1(b) for a pictorial representation of the reductions discussed so far. Given these connections and the existence of a 2-approximation for vertex cover, FVS, and PFDS, we are led to the following questions.

► **Question 1.** *What is the approximability of ρ -GRAPHDD, MATROIDFVS, and ρ -SUPMODDD? Do these admit constant factor approximations?*

1.1 Results

In this section, we give an overview of our technical results that resolve Question 1 up to a constant factor gap.

1.1.1 Connections between SubmodCover and SupmodDD

We obtain a logarithmic approximation for ρ -GRAPHDD, MATROIDFVS, and ρ -SUPMODDD via a reduction to the submodular cover problem and using the Greedy algorithm for it due to Wolsey [31]. First, we recall the submodular cover problem.

► **Definition 4** (SUBMODCOVER). *The submodular cover problem, denoted SUBMODCOVER, is defined as follows:*

Input: Integer-valued normalized non-decreasing submodular function $h : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ via evaluation oracle and element costs $c : V \rightarrow \mathbb{R}_{\geq 0}$

Goal: $\arg \min \{ \sum_{e \in F} c_e : F \subseteq V \text{ and } h(F) \geq h(V) \}$.

For a function $f : 2^V \rightarrow \mathbb{R}$, we define the marginal $f(v|S) := f(S + v) - f(S)$ for every $v \in V$ and $S \subseteq V$. We show the following result.

► **Theorem 5.** *Let $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ be an integer-valued normalized supermodular function and ρ be a rational number. Then, there exists a normalized non-decreasing submodular function $h : 2^V \rightarrow \mathbb{R}_{\geq 0}$ such that for $F \subseteq V$, we have that $\lambda_{f|_{V-F}}^* \leq \rho$ if and only if $h(F) \geq h(V)$. Moreover,*

1. *if ρ is an integer, then h is integer-valued,*
2. *$h(v) \leq \max\{0, f(v|V - v) - \rho\}$ for all $v \in V$, and*
3. *evaluation queries for the function h can be answered in polynomial time by making polynomial number of evaluation queries to the function f .*

We discuss the consequences of Theorem 5 for ρ -SUPMODDD. We recall that SUBMODCOVER admits a $(1 + \ln(\max_v h(v)))$ -approximation for input function h via the Greedy algorithm of Wolsey [31]. Consider ρ -SUPMODDD for integer-valued ρ . By Theorem 5, we have a reduction to SUBMODCOVER and consequently, we have a $(1 + \ln(\max_{v \in V} f(v|V - v)))$ -approximation. In particular, we note that ρ -GRAPHDD for integer-valued ρ and MATROIDFVS admit $O(\log n)$ -approximation, where n is the number of vertices in the input graph.

► **Corollary 6.** *ρ -SUPMODDD for integer-valued ρ admits an $(1 + \ln(\max_{v \in V} f(v|V - v)))$ -approximation, where $f : V \rightarrow \mathbb{Z}_{\geq 0}$ is the input integer-valued, normalized supermodular function. Consequently, ρ -GRAPHDD for integer valued ρ and MATROIDFVS admit $O(\log n)$ -approximations, where n is the number of vertices in the input graph.*

► **Remark 7.** The reduction from SUPMODDD to SUBMODCOVER is in some sense implicit in prior literature (see [18, 29] for certain special cases of supermodular functions). We note that the reduction from FVS to SUBMODCOVER which follows from this connection does not seem to be well-known in the literature, and the authors of this paper were not aware of it until recently.

From a structural point of view we also prove that SUBMODCOVER reduces to 1-SUPMODDD, thus essentially showing the equivalence of SUBMODCOVER and SUPMODDD. We believe that it is useful to have this equivalence explicitly known given that vertex deletion problems arise naturally but seem different from covering problems on first glance.

► **Theorem 8.** *Let $h : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ be an integer-valued normalized non-decreasing submodular function. Then, there exists a normalized supermodular function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ such that for $F \subseteq V$, we have that $h(F) \geq h(V)$ if and only if $\lambda_{f|_{V-F}}^* \leq 1$. Moreover,*

1. $f(v|V-v) = h(v) + 1$ for all $v \in V$, and
2. evaluation queries for the function f can be answered in polynomial time by making a constant number of evaluation queries to the function h .

1.1.2 Hardness of Approximation

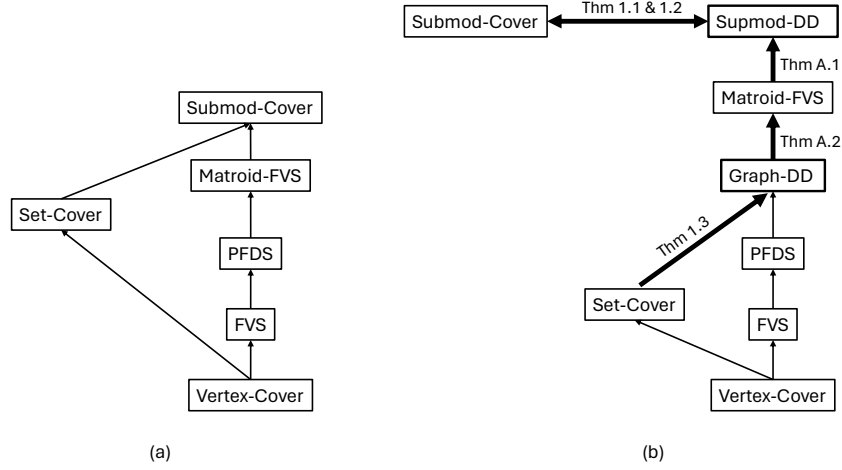
A starting point for our attempt to answer Question 1 was our belief that ρ -GRAPHDD for integer ρ admits a $(\rho + 1)$ -approximation via the primal-dual approach suggested by Fujito for MATROIDFVS [18]. This belief stems from Fujito’s work which showed a 2-approximation for vertex cover, FVS, PFDS, and MATROIDFVS for “uniformly sparse” matroids and our reduction showing that ρ -GRAPHDD for integral ρ is a special case of MATROIDFVS (see Appendix A of the full version of this work [6]). We note that the matroid that arises in the reduction is not a “uniformly sparse” matroid but has lot of similarities with it, so our initial belief was that a more careful analysis would lead to a constant factor approximation. However, to our surprise, after several unsuccessful attempts to prove a constant factor upper bound, we were able to show that for every integer $\rho \geq 2$, ρ -GRAPHDD is $\Omega(\log n)$ -hard to approximate via a reduction from Set Cover.

► **Theorem 9.** *For every integer $\rho \geq 2$, there is no $o(\log n)$ approximation for ρ -GRAPHDD assuming $P \neq NP$, where n is the number of vertices in the input instance.*

Thus, ρ -GRAPHDD exhibits a *phase transition*: it admits a 2-approximation for $\rho \leq 1$ (via Fujito’s results [18]) and becomes $\Omega(\log n)$ -hard for every integer $\rho \geq 2$. To conclude our hardness results, we note that since GRAPHDD is a special case of MATROIDFVS, which itself is a special case of SUPMODDD, both MATROIDFVS and SUPMODDD are $\Omega(\log n)$ -inapproximable. However, both these problems are also $O(\log n)$ -approximable via Corollary 6. Thus, we resolve the approximability of all these problems to within a small constant factor. We refer the reader to Figure 1 for an illustration of problems considered in this work and approximation-factor preserving reductions between them.

1.1.3 Bicriteria Approximations

The hardness result for 2-GRAPHDD (and ρ -GRAPHDD) motivates us to consider bicriteria approximation algorithms. Can we obtain constant factor approximation by relaxing the requirement of meeting the density target ρ exactly? We show that this is indeed possible. We consider an orientation based LP that was used recently to obtain a polynomial-time solvable LP to approximate FVS and PFDS [5]. We observed that this LP has an $\Omega(n)$ integrality gap when considering 2-GRAPHDD. Nevertheless, the LP is useful in obtaining the following bicriteria approximation.



■ **Figure 1** Reductions between problems of interest to this work. Arrow from Problem A to Problem B implies that Problem A has an approximation-preserving reduction to Problem B. Figure (a) consists of the connections between problems known prior to our work. Figure (b) showcases our results.

► **Theorem 10.** *There exists a polynomial time algorithm which takes as input a graph $G = (V, E)$, vertex deletion costs $c : V \rightarrow \mathbb{R}_{\geq 0}$, a target density $\rho \in \mathbb{R}$, and an error parameter $\epsilon \in (0, 1/2)$, and returns a set $S \subseteq V$ such that:*

1. $\lambda_{G-S}^* \leq \left(\frac{1}{1-2\epsilon}\right) \cdot \rho$,
2. $\sum_{u \in S} c_u \leq \left(\frac{1}{\epsilon}\right) \cdot \text{OPT}$,

where OPT denotes the cost of an optimum solution to ρ -GRAPHDD on the instance (G, c) .

Next, we consider ρ -SUPMODDD. Unlike the case of graphs, it is not clear how to write an integer programming formulation for SUPMODDD whose LP-relaxation is polynomial-time solvable. Instead, we take inspiration from the very recent work of [32] on vertex deletion to reduce treewidth. We design a combinatorial randomized algorithm that yields a bicriteria approximation for SUPMODDD, where the bicriteria approximation bounds are based on a parameter c_f that depends on the input supermodular function f . For a normalized supermodular function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$, we define

$$c_f := \max \left\{ \frac{\sum_{u \in S} f(u|S-u)}{f(S)} : S \subseteq V \right\}.$$

This parameter c_f was defined in a recent work on DSS to unify the analysis of the greedy peeling algorithm for DSG [9]. We note that $1 \leq c_f \leq |V|$ and moreover, $c_f = 1$ if and only if the function f is modular. If f is the induced edge function of a graph (i.e., $f(S)$ is the number of edges with all its end-vertices in S for every subset S of vertices), then $c_f \leq 2$. This follows from the observation that the sum of degrees is at most twice the number of edges in a graph. Similarly, if f is the induced edge function of a hypergraph with rank r (i.e., all hyperedges have size at most r), then $c_f \leq r$. We show the following bicriteria approximation for SUPMODDD.

► **Theorem 11.** *There exists a randomized polynomial time algorithm which takes as input a normalized monotone supermodular function $f : 2^V \rightarrow \mathbb{Z}$ (given by oracle access), element deletion costs $c : V \rightarrow \mathbb{R}_{\geq 0}$, a target density $\rho \in \mathbb{R}$, and an error parameter $\epsilon \in (0, 1)$, and returns a set $S \subseteq V$ such that:*

1. $\lambda_{f|_{V-S}}^* \leq c_f(1 + \epsilon) \cdot \rho$, and
2. $\mathbb{E} \left[\sum_{u \in S} c_u \right] \leq c_f \left(1 + \frac{1}{\epsilon} \right) \cdot \text{OPT}$,

where OPT denotes the cost of an optimum solution to ρ -SUPMODDD on the instance (f, c) .

As a consequence of Theorem 11, we obtain a bicriteria approximation for density deletion problems in graphs and r -rank hypergraphs. We note that the bicriteria guarantee that we get from this theorem for graphs is weaker than the guarantee stated in Theorem 10. We discuss another special case of SUPMODDD where the supermodular function of interest has bounded c_f to illustrate the significance of Theorem 11. Given a graph $G = (V, E)$ and a parameter $p \geq 1$, the p -mean density of G is defined as $\max\{\sum_{u \in S} d_S(u)^p / |S| : S \subseteq V\}$, where $d_S(u)$ is the number of edges in $E[S]$ incident to the vertex u . The p -mean density of graphs was introduced and studied by Veldt, Benson, and Kleinberg [30]. Subsequent work by Chekuri, Quanrud, and Torres [9] showed that p -mean density is a special case of the densest supermodular set problem (i.e., DSS) where the supermodular function $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$ of interest is given by $f(S) := \sum_{u \in S} d_S(u)^p$ for every $S \subseteq V$ and moreover $c_f \leq (p + 1)^p$. We note that the natural vertex deletion problem is the p -mean density deletion problem: given a graph $G = (V, E)$ with vertex deletion costs, find a min-cost subset of vertices to delete so that the p -mean density of the remaining graph is at most a given threshold. Since $c_f \leq (p + 1)^p$ for the function f of interest here, Theorem 11 implies a bicriteria approximation for p -mean density deletion for integer-valued p . An interesting open question is to obtain better bicriteria approximation for ρ -SUPMODDD – in particular, can we remove the dependence on c_f ?

Organization. In Section 2, we show an approximation-preserving reduction from SETCOVER to ρ -GRAPHDD and prove Theorem 9. In Section 3, we give a bicriteria approximation for ρ -SUPMODDD and prove Theorem 11. Finally, in Section 4 we show the connections between SUPMODDD and SUBMODULAR COVER by proving Theorems 5 and 8. We defer the proof of Theorem 10 to the full version of this work [6].

2 Hardness of Approximation

In this section, we show Theorem 9, i.e., we show an approximation preserving reduction from SETCOVER to ρ -GRAPHDD. We recall the set cover problem and its inapproximability.

► **Definition 12** (SETCOVER). *The set cover problem, denoted SETCOVER, is defined as follows:*

Input: Finite Universe \mathcal{U} , Family $\mathcal{S} \subseteq 2^{\mathcal{U}}$ with costs $c : \mathcal{S} \rightarrow \mathbb{R}_{\geq 0}$
Goal: $\arg \min \{ \sum_{S \in \mathcal{F}} c_S : \mathcal{F} \subseteq \mathcal{S} \text{ with } \cup_{S \in \mathcal{F}} S = \mathcal{U} \}$.

► **Theorem 13** ([14, 15]). *For every $\epsilon > 0$, there does not exist a $(1 - \epsilon) \ln n$ -approximation for SETCOVER assuming $P \neq NP$, where n is the size of the input instance.*

We will also need the following characterization of density via *orientations*. We recall that an *orientation* of a graph $G = (V, E)$ assigns each edge $\{u, v\}$ to either u or v . For notational convenience, we use $\vec{G} = (V, \vec{E})$ to denote an orientation of the graph G and $d_{\vec{G}}^{\text{in}}(u)$ to denote the indegree of a vertex $u \in V$ (i.e., the number of edges assigned to u) in the oriented graph \vec{G} . The following connection between density and orientations will be used in our hardness reduction. This characterization is implied by a result of Hakimi [16] and also by the dual of Charikar’s LP to solve DSG [7].

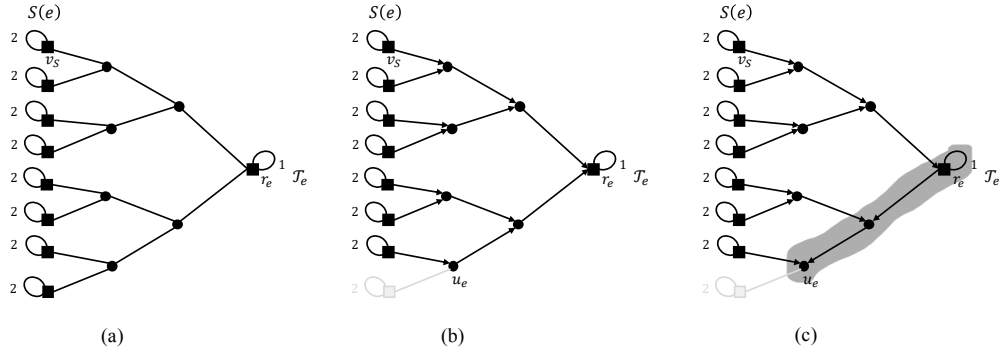


Figure 2 The figure in (a) depicts the subgraph of the construction corresponding to element $e \in \mathcal{U}$. Here, $f = 8$. The figure in (b) depicts the intermediate orientation \vec{H} for the subgraph of H corresponding to an element $e \in \mathcal{U}$. The greyed-out set-vertex at the bottom represents that this vertex is in X_F . The figure in (c) depicts the final orientation for the subgraph from the figure in (b) after reorientation. The highlighted edges are those that have been reoriented.

► **Proposition 14.** Let $G = (V, E)$ be a graph and let $\rho \in \mathbb{Z}_{\geq 0}$ be an integer value. Then, we have that $\lambda_G^* \leq \rho$ if and only if there exists an orientation $\vec{G} = (V, \vec{E})$ of the graph G such that $d_G^{in}(u) \leq \rho$ for every $u \in V$.

We now restate and prove Theorem 9.

► **Theorem 9.** For every integer $\rho \geq 2$, there is no $o(\log n)$ approximation for ρ -GRAPHDD assuming $P \neq NP$, where n is the number of vertices in the input instance.

Proof. We show the theorem when the target density $\rho = 2$ via a reduction from SETCOVER. At the end of the proof, we remark on how to modify the reduction to obtain hardness for all integers $\rho \geq 2$ as claimed in the theorem. Let $(\mathcal{S}, \mathcal{U}, c : \mathcal{S} \rightarrow \mathbb{R})$ be a SETCOVER instance. We will assume (without loss of generality) that all elements in \mathcal{U} have the same frequency $f \geq 4$ which is a power of 2 – we note that this assumption is not a technical requirement and is only for ease of exposition. For this instance, we construct an instance $(G = (V, E), c_G : V \rightarrow \mathbb{R})$ of 2-GRAPHDD as follows:

1. *Add vertices representing sets:* For each set $S \in \mathcal{S}$, add a set-vertex v_S to V .
2. *Add binary trees representing elements:* For each element $e \in \mathcal{U}$, add a complete binary tree with the f leaves as the set-vertices corresponding to the sets containing e . We denote this tree as \mathcal{T}_e and its root as r_e .
3. *Add self-loops:* For each element $e \in \mathcal{U}$, add a self-loop to the root vertex r_e of the tree \mathcal{T}_e . For each set $S \in \mathcal{S}$, add $\rho = 2$ self-loops to the vertex v_S .
4. *Define cost function:* We define the cost function $c_G : V \rightarrow \mathbb{R}$ as follows: $c_G(v_S) = c(S)$ for all $S \in \mathcal{S}$ and $c_G(u) = \infty$ for all $u \in V - \{v_S : S \in \mathcal{S}\}$.

We refer the reader to Figure 2(a) for pictorial depictions of the instance constructed via the reduction above. The next claim shows the correctness of our reduction and also implies that our reduction is approximation-preserving. The approximation hardness guarantee of the theorem when the target density $\rho = 2$ then follows by Theorem 13 and the observation that the number of vertices $|V|$ is a constant factor of the size of the input set cover instance.

▷ **Claim 15.** The instance (G, c_G) has a feasible solution to 2-GRAPHDD of finite cost T if and only if $(\mathcal{S}, \mathcal{U}, c)$ has a SETCOVER of cost T .

Proof. We first show the forward direction of the claim. Let $X \subseteq V$ be a feasible solution to 2-GRAPHDD of finite cost T . By construction, we have that $X \subseteq \{v_S : S \in \mathcal{S}\}$. Let $F := \{S : v_S \in X\}$ denote the corresponding sets. By way of contradiction, suppose F is not a set cover. Consequently, there exists an element $e \in \mathcal{U}$ not covered by F . For convenience, we use $\mathcal{S}(e) := \{S \in \mathcal{S} : e \in S\}$ to denote the sets that contain the element e . We note that since the element e is not covered by F , we have that $\{v_S : S \in \mathcal{S}(e)\} \cap X = \emptyset$. Let V_e denote the set of vertices obtained by including all vertices of the binary tree \mathcal{T}_e . Then, the following gives us a contradiction:

$$2 \geq \lambda_{G-X}^* \geq \frac{|E[V_e]|}{|V_e|} = \frac{(2f+1) + (2f-2)}{2f-1} > 2.$$

Here, the first inequality is because X is a feasible solution to 2-GRAPHDD. The second inequality is by definition of graph density. The equality is because there are $(2f+1)$ self loop edges, $(2f-2)$ non-self loop edges, and $2f-1$ vertices in the tree \mathcal{T}_e by construction.

We now show the reverse direction. Let $F \subseteq \mathcal{S}$ be a set cover. Let $X_F = \{v_S : S \in F\}$. Then, we show that the graph $H := G - X_F$ has density at most 2. By Proposition 14(2), it suffices to exhibit an orientation of the graph H in which the indegree of every vertex is at most 2. We first consider the following intermediate orientation of G . For each element $e \in \mathcal{U}$, we do the following: for vertex $u \in \mathcal{T}_e - \{r_e\}$, we denote $p(u)$ as the (unique) parent of u in the (rooted) tree, and we orient the edge $(u, p(u))$ towards the parent $p(u)$. All self-loops are assumed to be trivially oriented. Let $\vec{H} := (V_H, \vec{E}_H)$ denote this intermediate orientation restricted to the graph H , and $\mathcal{R} := \{r_e : e \in \mathcal{U}\}$ denote the set of all root vertices. Refer to Figure 2(b) for a pictorial depiction of orientation \vec{H} . We now make three important observations regarding the indegrees in the orientation \vec{H} .

► **Observation 16.** *We have the following:*

1. for all $u \in V_H - \mathcal{R}$, $d_{\vec{H}}^{\text{in}}(u) \leq 2$,
2. for all $r \in \mathcal{R}$, $d_{\vec{H}}^{\text{in}}(r) = 3$, and
3. for each element $e \in \mathcal{U}$, there exists a set $S_e \in \mathcal{S}(e)$ such that $d_{\vec{H}}^{\text{in}}(p(v_{S_e})) \leq 1$.

Proof. We show each statement separately below.

1. We note that the statement easily follows by construction for the set vertices v_S . Let $e \in \mathcal{U}$ be an arbitrary element and let $u \in \mathcal{T}_e - (\{v_S : S \in \mathcal{S}(e)\} \cup \{r_e\})$ be a non-root internal vertex of the binary tree. Since u has exactly two child nodes and one parent node in \mathcal{T}_e , we have that $d_{\vec{H}}^{\text{in}}(u) \leq 2$. We note that the inequality may be strict if any children of u belong to the set X_F .
2. Let $e \in \mathcal{U}$. We note that r_e has exactly 2 children and 1 self loop, and consequently $d_{\vec{H}}^{\text{in}}(r_e) \leq 3$ as claimed. Here, we note that both child of r_e belongs to the set X_F because of our simplifying assumption that $f \geq 4$.
3. Let $e \in \mathcal{U}$. Since F is a set cover, there exists a set $S \in \mathcal{S}(e)$ such that $S \in F$. Consequently, $v_S \in X_F$, and so $d_{\vec{H}}^{\text{in}}(p(v_S)) \leq 1$ by construction. ◀

We now use the orientation \vec{H} and Observation 16 to construct the orientation which certifies that the graph H has density at most 2. We note that by Observation 16(1) and (2), it suffices to modify the orientation \vec{H} to reduce the indegree of all root vertices in \mathcal{R} to 2 while keeping all other indegrees at most 2. Consider an arbitrary element $e \in \mathcal{U}$. Let u_e be an arbitrary vertex of \mathcal{T}_e such that $d_{\vec{H}}^{\text{in}}(u_e) \leq 1$. We note that such a vertex exists by Observation 16(3). We consider the unique path P_e from u_e to r_e in \mathcal{T}_e . We note that by the construction of the orientation \vec{H} , every edge along this path is oriented in the direction of the path. Consider the orientation obtained by reorienting these edges in the reverse

direction of the path. Refer to Figure 2(c) for a pictorial depiction of the modified orientation. We note that only the indegrees of r_e and u_e change due to this reorientation. In particular, for all $e \in \mathcal{U}$, we have that $d_{\vec{H}}^{\text{in}}(u_e) \leq 2$ and $d_{\vec{H}}^{\text{in}}(r_e) = 2$. This concludes the proof. \triangleleft

The preceding reduction can be modified to show approximation hardness for all integral $\rho \geq 2$. Suppose that $\rho = 2 + \alpha$, where $\alpha \in \mathbb{Z}_{\geq 0}$. Then, we construct the same graph as above with α additional self-loops on each vertex. The proof generalizes. \blacktriangleleft

► **Remark 17.** The addition of self-loops is not technically necessary for the construction; they simply make it more straightforward. Specifically, a vertex u with $\gamma \in \mathbb{Q}_{\geq 1}$ self-loops can be replaced by a subgraph with density exactly γ . In this subgraph, one vertex, say h_u , is identified with u and its cost is defined as $c(h_u) := c(u)$. All other vertices have infinite cost. All edges incident to u are then redirected to connect to h_u instead.

3 Bicriteria Approximation for GraphDD

In this section, we describe a randomized combinatorial bicriteria approximation algorithm for ρ -SUPMODDD and prove Theorem 11. The algorithm is inspired by the ideas of the recent work of Włodarczyk [32]. Our algorithm is based on the following idea. Suppose that we had non-negative *potentials* $\pi : V \rightarrow \mathbb{R}_{\geq 0}$ for the elements of the ground set such that the potential value $\sum_{u \in X} \pi(u)$ of an optimal solution $X \subseteq V$ is large, say at least $\alpha \cdot \sum_{u \in V} \pi(u)$. Then, a natural algorithm – at least when the vertex deletion costs are uniform – would be to compute the potentials, sample an element in proportion to the potentials and delete it, define a residual instance, and repeat. This would ensure an α -approximation for the problem in expectation via a martingale argument.

Unfortunately, our hardness result suggests that we are unlikely to obtain good potentials. In fact, the hard instances seem to be the functions that have density very close to the target density. However, we leverage supermodularity to show that if the density of the input function is at least β times the target density, then we can indeed find such good vertex potentials. In order to ensure that the density of the input function is at least β times the target density, we perform a preprocessing step to prune certain elements from the ground set without changing the cost of an optimal solution. Overall, this gives us an (α, β) -bicriteria guarantee, where the values α and β are as given in Theorem 11. We also note that the cost function to delete vertices may be arbitrary – we overcome this by using the natural bang-per-buck sampling strategy, i.e. we sample a vertex u in proportion to $\pi(u)/c(u)$.

The rest of the section is organized as follows. In Section 3.1 we give our preprocessing step based on the dense decomposition of supermodular functions. In Section 3.2 we describe our element potentials based on marginal gains of supermodular functions. In Section 3.3 we present our algorithm and complete the proof of Theorem 11.

3.1 Preprocessing via Dense Decomposition

We discuss the *dense decomposition* of a normalized non-negative supermodular set function [17, 20] and prove a lemma that will enable us to use it as a preprocessing step.

► **Definition 18** ([20]). *Let $f : 2^V \rightarrow \mathbb{R}_+$ be a non-negative normalized supermodular function. A sequence $(V_1, \rho_1), (V_2, \rho_2), \dots, (V_k, \rho_k)$ is the dense decomposition of f if*

1. V_1, \dots, V_k is a partition of V obtained iteratively as follows: for $i = 1, 2, \dots, k$, V_i is the inclusion-wise maximal set $S \subseteq V - \bigcup_{j \in [i-1]} V_j$ that maximizes

$$\frac{f\left(S \cup \bigcup_{j \in [i-1]} V_j\right) - f\left(\bigcup_{j \in [i-1]} V_j\right)}{|S|}$$

2. the values ρ_1, \dots, ρ_k are obtained as

$$\rho_i := \frac{f\left(\bigcup_{j \in [i]} V_j\right) - f\left(\bigcup_{j \in [i-1]} V_j\right)}{|V_i|} \quad \forall i \in [k].$$

We note that the dense decomposition can also be viewed algorithmically as the output of a recursive process which computes the unique inclusion-wise maximal set S that maximizes the ratio $f(S)/|S|$ and recurses on the *contracted* function $f_{/S} : 2^{V-S} \rightarrow \mathbb{R}$ defined as $f_{/S}(X) := f(S \cup X)$ for all $X \subseteq V - S$. It can be shown that this decomposition is unique for a supermodular f . The following lemma allows us to use the dense decomposition as an algorithmic preprocessing step in the next section. In particular, the lemma says that the dense decomposition can be used to find a set $R \subseteq V$ such that it suffices to focus on solving the ρ -SUPMODDD problem on the function restriction $f|_R$; and additionally, the ground set elements of the restricted function have *large* marginal gains.

► **Lemma 19.** *Let $f : 2^V \rightarrow \mathbb{R}_{\geq 0}$ be a normalized non-negative supermodular function, $c : V \rightarrow \mathbb{R}_+$ be a cost function, and $\rho \in \mathbb{R}_+$ be a positive real value. Moreover, let $(V_1, \phi_1), (V_2, \phi_2), \dots, (V_k, \phi_k)$ be the dense decomposition of (f, V) and for $\rho' > \rho$, let $R := \bigcup_{i \in [k]: \phi_i > \rho'} V_i$. Then, we have that*

1. every feasible solution to ρ' -SUPMODDD for the function $f|_R$ is also a feasible solution to ρ' -SUPMODDD for the function f ,
2. $\text{OPT}(f) \geq \text{OPT}(f|_R)$, where $\text{OPT}(f)$ and $\text{OPT}(f|_R)$ denote the costs of optimal solutions to ρ -SUPMODDD for the functions f and $f|_R$ respectively,
3. $f(v|R - v) \geq \rho'$ for all $v \in R$, and
4. the set R can be computed in polynomial time given access to a function evaluation oracle for f .

Proof. We show all four properties separately below.

1. Let $X \subseteq R$ be a feasible solution to ρ' -SUPMODDD for the function $f|_R$ and by way of contradiction suppose that X is not a feasible solution to ρ' -SUPMODDD for the function f . Thus, there exists a set $S \subseteq V - X$ such that $f(S)/|S| > \rho'$. We note that this set S cannot be contained in $R - X$ since otherwise X would not be a feasible solution to ρ' -SUPMODDD for $f|_R$. Then, the following gives us the required contradiction:

$$\begin{aligned} \rho' < \frac{f(S)}{|S|} &\leq \frac{f(S \cup R) - f(R) + f(S \cap R)}{|S - R| + |S \cap R|} \\ &\leq \max \left\{ \frac{f(S \cup R) - f(R)}{|S - R|}, \frac{f(S \cap R)}{|S \cap R|} \right\} \\ &\leq \max \left\{ \max \left\{ \frac{f(R \cup S') - f(R)}{|S'|} : S' \subseteq V - R \right\}, \lambda_{f|_{R-X}}^* \right\} \\ &\leq \rho'. \end{aligned}$$

Here, the second inequality is by supermodularity of the function f . The third inequality is by the observation that $(a+b)/(c+d) \leq \max\{a/c, b/d\}$ for non-negative numbers a, b, c, d . For the final inequality, observe that $\lambda_{f|_{R-X}}^* \leq \rho'$ because X is a feasible ρ' -SUPMODDD for $f|_R$. Furthermore, we have that $\max \left\{ \frac{f(R \cup S') - f(R)}{|S'|} : S' \subseteq V - R \right\} \leq \rho'$ by definition of the dense decomposition and R .

2. Let $X \subseteq V$ be an optimal ρ -SUPMODDD for f w.r.t. cost function c . Then, we note that $X \cap R$ is a feasible ρ -SUPMODDD for $f|_R$. This can be easily observed as follows: by way of contradiction, suppose that $X \cap R$ is not a feasible ρ -SUPMODDD for $f|_R$.

43:12 On Deleting Vertices to Reduce Density in Graphs and Supermodular Functions

Then, there exists a set $S \subseteq R - X$ such that $f|_R(S)/|S| > \rho$. Consequently, we have that $\lambda_{f|_{V-X}}^* \geq f(S)/|S| = f|_R(S)/|S| > \rho$, a contradiction to X being a feasible ρ -SUPMODDD for f . Then, $\text{OPT}(f) \geq \text{OPT}(f|_R)$ follows by non-negativity of c .

3. By way of contradiction, suppose that there exists a vertex $v \in R$ such that $f(v|R-v) \leq \rho'$. We recall that $R = \bigcup_{i \in [k]: \phi_i > \rho'} V_i$. Let $j \in [k]$ be such that $v \in V_j$. For convenience, we will let $U_{j-1} := \bigcup_{i \in [j-1]} V_i$ and $U_j = \bigcup_{i \in [j]} V_i$. We note that U_{j-1} and U_j are contained in R and $\rho_j := \frac{f(U_j) - f(U_{j-1})}{|V_j|}$ by definition of the dense decomposition. Moreover, $\rho_j > \rho'$ by the definition of the set R , and so by supermodularity we have that $f(v|U_j - v) \leq f(v|R - v) \leq \rho' < \rho_j$. Then, the following sequence of inequalities gives us the required contradiction.

$$\begin{aligned}
 \rho_j &= \frac{f(U_j) - f(U_{j-1})}{|V_j|} \\
 &= \frac{f(U_j) - f(U_j - v) + f(U_j - v) - f(U_{j-1})}{|V_j|} \\
 &= \frac{f(v|U_j - v) + f(U_j - v) - f(U_{j-1})}{1 + (|V_j| - 1)} \\
 &< \frac{\rho_j + f(U_j - v) - f(U_{j-1})}{1 + (|V_j| - 1)} \\
 &\leq \max \left\{ \rho_j, \frac{f(U_j - v) - f(U_{j-1})}{|V_j| - 1} \right\} \\
 &\leq \max \left\{ \rho_j, \max \left\{ \frac{f(U_{j-1} \cup S) - f(U_{j-1})}{|S|} : S \subseteq V - U_{j-1} \right\} \right\} \\
 &\leq \rho_j,
 \end{aligned}$$

where the second inequality is by the observation that $(a + b)/(c + d) \leq \max\{a/c, b/d\}$ for non-negative numbers a, b, c, d . Here, we note that $|V_j| \geq 2$ since otherwise, $V_j = \{v\}$ and so by supermodularity we have that $\rho_j = f(U_{j-1} + v) - f(U_{j-1}) = f(v|U_{j-1}) \leq f(v|R - v) < \rho_j$, a contradiction.

4. It is well-known that the dense decomposition (and consequently the set R) can be computed in polynomial time (given access to the function evaluation oracle for f) via supermodular maximization. This is implicit in Fujishige's work on principle partitions [17], and is also explicitly considered in more recent works on dense decompositions for supermodular functions [20]. We omit the details of a formal proof here for brevity. ◀

3.2 Element Potentials via Marginal Gains

We now show that the marginal gains of elements relative to the entire ground set are good potentials for a sampling-based algorithm for ρ -SUPMODDD when all the marginal gains of the input function are large enough, in particular, at least $c_f(1 + \epsilon)\rho$.

► **Lemma 20.** *Let $\rho \in \mathbb{R}$ and $\epsilon \in (0, 1)$. Let $f : 2^V \rightarrow \mathbb{Z}$ be a normalized monotone supermodular function such that $f(u|V - u) \geq c_f(1 + \epsilon)\rho$ for all $u \in V$, and let $X \subseteq V$ be a ρ -SUPMODDD for f . Then, we have that*

$$\sum_{u \in X} f(u|V - u) \geq \frac{1}{c_f(1 + 1/\epsilon)} \sum_{u \in V} f(u|V - u).$$

Proof. By supermodularity of f , we have that

$$\begin{aligned} \sum_{u \in X} f(V - u) &\leq (|X| - 1)f(V) + f(V - X) \text{ and hence,} \\ \sum_{u \in X} f(u|V - u) &\geq f(V) - f(V - X). \end{aligned} \quad (1)$$

By way of contradiction, suppose that the lemma is false. Then, we have the following.

$$\begin{aligned} \sum_{u \in V} f(u|V - u) &\leq c_f f(V) \\ &= c_f(f(V) - f(V - X) + f(V - X)) \\ &\leq c_f(f(V) - f(V - X)) + c_f \rho |V - X| \\ &\leq c_f \sum_{u \in X} f(u|V - u) + c_f \rho |V - X| \\ &< \frac{1}{1 + 1/\epsilon} \sum_{u \in V} f(u|V - u) + c_f \rho |V - X|. \end{aligned}$$

Here, the first inequality is by definition of the parameter c_f . The second inequality is because X is a feasible ρ -SUPMODDD for the function f . The third inequality is by (1). The final inequality is by our contradiction assumption. Then, on rearranging the terms, we obtain the following contradiction.

$$c_f \rho |V - X| > \left(1 - \frac{1}{1 + 1/\epsilon}\right) \sum_{u \in V} f(u|V - u) = \frac{1}{(1 + \epsilon)} \sum_{u \in V} f(u|V - u) \geq c_f \rho |V|,$$

where the final inequality is because $f(u|V - u) \geq c_f(1 + \epsilon)\rho$ for all $u \in V$. \blacktriangleleft

3.3 Random Deletion Algorithm

We now describe our bicriteria algorithm for ρ -SUPMODDD and analyze its approximation factor. Algorithm 1, Lemma 23 and Lemma 24 together complete the proof of Theorem 11.

Algorithm. Our algorithm takes as input (1) a normalized non-negative supermodular function $f : 2^V \rightarrow \mathbb{R}_+$, (2) element deletion costs $c : V \rightarrow \mathbb{R}_+$, (3) target density $\rho \in \mathbb{R}_+$, and (4) error parameter $\epsilon > 0$. The algorithm returns a set $S \subseteq V$ which starts off as the empty-set and is then constructed element-by-element. This is done iteratively as follows. Let $\beta := c_f(1 + \epsilon)$. If the function f has density at most $\beta\rho$, then the algorithm breaks and returns the current set S . Otherwise, the algorithm first computes the dense decomposition $(V_1, \phi_1), (V_2, \phi_2), \dots, (V_k, \phi_k)$ of the function f , defines the set $R := \cup_{i \in [k]: \phi_i > \beta\rho} V_i$, and redefines the function f to be the restricted function $f|_R : 2^R \rightarrow \mathbb{R}_{\geq 0}$ – we use `DENSEDECOMPOSITIONPREPROCESS(f, ρ)` to denote a subroutine that computes the set R and returns the tuple $(f|_R, R)$. Next, the algorithm samples a random element u from the (modified) set V in proportion to the ratio $f(u|V - u)/c(u)$. The algorithm then adds the vertex u to the set S , restricts f to the ground set $V - u$, and repeats the previous steps. We give a formal description of the algorithm in Algorithm 1.

Martingales. For the analysis of our randomized algorithm, we will require the following concepts from probability theory.

■ **Algorithm 1** Bicriteria approximation algorithm for ρ -SUPMODDD.

ALGORITHM $\left((f : 2^V \rightarrow \mathbb{R}, c), \rho, \epsilon \right)$:

1. $S := \emptyset$
2. **while** $\lambda_f^* > c_f(1 + \epsilon)\rho$:
 - a. Redefine $(f, V) := \text{DENSEDECOMPOSITIONPREPROCESS}(f, c_f(1 + \epsilon)\rho)$
 - b. $u :=$ vertex sampled from V according to the following distribution:

$$\Pr(u = v) := \frac{f(v|V - v)}{c(v) \cdot W} \quad \forall v \in V, \text{ where } W := \sum_{v \in V} \frac{f(v|V - v)}{c(v)} \text{ is a normalizing factor}$$

- c. $S := S + u$ and $f := f|_{V-u}$
 3. **return** S .
-

► **Definition 21.**

1. A sequence of random variables P_1, P_2, \dots is called a supermartingale w.r.t. the sequence X_1, X_2, \dots of random variables if for each $i \in \mathbb{Z}_+$ it holds that (i) P_i is a function of X_1, \dots, X_i , (ii) $\mathbb{E}[|P_i|] < \infty$ and (iii) $\mathbb{E}[P_{i+1} | X_1, \dots, X_i] \leq P_i$.
2. A random variable T is called a stopping time with respect to the sequence of random variables P_1, P_2, \dots if for each $i \in \mathbb{Z}_+$, the event $(T \leq i)$ depends only on P_1, \dots, P_i .

The following result shows that the expected value of a random variable in the supermartingale process only decreases with time. This will be crucial in analyzing the performance of Algorithm 1.

► **Theorem 22** (Doob's Optional-Stopping Theorem). *Let P_0, P_1, \dots be a supermartingale w.r.t. the sequence X_1, X_2, \dots of random variables and ℓ be a stopping time with respect to the process P . Suppose that $\Pr(\ell \leq n) = 1$ for some integer $n \in \mathbb{Z}_+$. Then, we have that $\mathbb{E}[P_\ell] \leq \mathbb{E}[P_0]$.*

Algorithm Analysis. Henceforth, we consider the execution of Algorithm 1 on a fixed input instance (f, c, ρ, ϵ) . Let $\ell \in \mathbb{Z}_+$ be the number of iterations of the while-loop – we note that ℓ is a random variable with value at most n since at every iteration of the while-loop, the size of the ground set decreases by at least 1. Throughout the analysis, we will index the (random) variables at the i^{th} iteration of the algorithm with the subscript i for all $i \in [\ell]$. In particular, we let S_i denote the set S at the start of the i^{th} iteration (so $S_1 := \emptyset$, and S_{i+1} is defined by Step 2(c)), $f_i : 2^{V_i} \rightarrow \mathbb{R}_{\geq 0}$ denote the preprocessed function f after step 2(a), and u_i denote the sampled vertex u after step 2(b) of the i^{th} iteration of the algorithm. For simplicity, we define $S_j := S$, and f_j to be the empty-function for all $j \geq \ell$. The next lemma shows that the density of the function after deleting the set S is at most $c_f(1 + \epsilon)\rho$, i.e. S is a feasible solution to $(c_f(1 + \epsilon)\rho)$ -SUPMODDD for the function f . The proof easily follows by considering any fixed execution of the algorithm and leveraging Lemma 19(1) while inducting on ℓ . We omit details of the proof here for brevity.

► **Lemma 23** (Approximate Feasibility). $\lambda_{f|_{V-S}}^* \leq c_f(1 + \epsilon)\rho$.

The next lemma shows that the expected cost of the solution returned by the algorithm is at most $c_f(1 + 1/\epsilon)\rho$ times the cost of the optimal ρ -SUPMODDD of the function f . For any restriction g of the function f , we use $\text{OPT}(g)$ to denote the value of an optimal ρ -SUPMODDD for g with respect to the cost function c .

► **Lemma 24** (Approximate Cost). $\mathbb{E}[c(S)] \leq c_f(1 + 1/\epsilon)\text{OPT}(f)$.

Proof. For ease of exposition, we will use $\alpha := c_f(1 + 1/\epsilon)$. We consider the sequence of random variables P_1, P_2, \dots , where $P_i := c(S_i) + \alpha\text{OPT}(f_i)$ for all $i \in \mathbb{Z}_+$. Our strategy will be to first show that this sequence of random variables is a supermartingale, and then apply Doob's Optional-Stopping Theorem with stopping time n to bound the expected cost of the set returned by the algorithm (note that $\ell \leq n$ since with each iteration of the while-loop, the size of the ground set decreases by at least 1). Before showing that the sequence is a supermartingale, we first show that the expected cost of a vertex chosen in step 2(b) of an iteration of the algorithm is at most an α -fraction of the expected decrease in the optimum value during the iteration.

▷ **Claim 25.** $\mathbb{E}[c(u_i)|u_1, u_2, \dots, u_{i-1}] \leq \alpha\mathbb{E}[\text{OPT}(f_i) - \text{OPT}(f_{i+1})|u_1, u_2, \dots, u_{i-1}]$ for all $i \in [\ell]$.

Proof. Let X_i be an optimal ρ -SUPMODDD for f_i . We have the following:

$$\begin{aligned} \mathbb{E}[c(u_i)|u_1, u_2, \dots, u_{i-1}] &= \sum_{v \in V_i} \Pr(u_i = v) \cdot c(v) \\ &= \frac{1}{W} \sum_{v \in V_i} f_i(v|V_i) \\ &\leq \frac{\alpha}{W} \sum_{v \in X_i} f_i(v|V_i) \\ &= \alpha \sum_{v \in X_i} \Pr(u_i = v) \cdot c(v), \end{aligned}$$

where the first inequality is by Lemma 20 and the fact that $f(v|V_i) \geq c_f(1 + \epsilon)\rho$ by our preprocessing (Step 2(a)) and Lemma 19(3). We now show that because X_i is an optimal solution for f_i , the final expression in the above can be upper bounded by $\alpha\mathbb{E}[\text{OPT}(f_i) - \text{OPT}(f_{i+1})]$, thereby completing the proof of the claim. This can be seen as follows:

$$\begin{aligned} \mathbb{E}[\text{OPT}(f_i) - \text{OPT}(f_{i+1})] &= \sum_{v \in V_i} \mathbb{E}[\text{OPT}(f_i) - \text{OPT}(f_{i+1})|u_i = v] \cdot \Pr(u_i = v) \\ &\geq \sum_{v \in X_i} \mathbb{E}[\text{OPT}(f_i) - \text{OPT}(f_{i+1})|u_i = v] \cdot \Pr(u_i = v) \\ &\geq \sum_{v \in X_i} \mathbb{E}[\text{OPT}(f_i) - \text{OPT}(f_i|_{V_i-v})|u_i = v] \cdot \Pr(u_i = v) \\ &\geq \sum_{v \in X_i} \mathbb{E}[c(X_i) - c(X_i - v)|u_i = v] \cdot \Pr(u_i = v) \\ &= \sum_{v \in X_i} c(v) \cdot \Pr(u_i = v). \end{aligned}$$

Here, the second inequality is by Step 2(b) of Algorithm 1 which says that the function f_{i+1} is defined to be $\text{DENSEDECOMPOSITIONPREPROCESS}(f_i|_{V_i-v}, c_f(1 + \epsilon)\rho)$ and Lemma 19(2). The third inequality is because X_i is an optimal solution for f_i and $X_i - v$ is a feasible solution for $f_i|_{V_i-v}$. ◁

We now show that the sequence P_1, P_2, \dots is a supermartingale w.r.t. the sequence of random variables u_1, u_2, \dots chosen by the algorithm.

▷ **Claim 26.** The sequence of random variables P_1, P_2, \dots is a supermartingale w.r.t. the sequence of random variables u_1, u_2, \dots .

43:16 On Deleting Vertices to Reduce Density in Graphs and Supermodular Functions

Proof. Let $i \in \mathbb{Z}_+$ be arbitrary. We note that P_i has finite expectation and also is fully determined by the subsequence u_1, \dots, u_i . Thus, our goal is to show that $\mathbb{E}[P_{i+1}|u_1, \dots, u_i] \leq P_i$. This is equivalent to showing $\mathbb{E}[P_{i+1} - P_i|u_1, \dots, u_i] \leq 0$. We note that this inequality indeed holds because

$$\mathbb{E}[P_{i+1} - P_i|u_1, \dots, u_i] = \mathbb{E}\left[c(u_{i+1}) - \alpha(\text{OPT}(f_i) - \text{OPT}(f_{i+1}))|u_1, \dots, u_i\right] \leq 0,$$

where the inequality is by Claim 25. \triangleleft

By Claim 26, the sequence P_1, P_2, \dots is a supermartingale w.r.t. the sequence u_1, u_2, \dots of random variables. Consider the stopping time ℓ . By Theorem 22, we have that $\mathbb{E}[P_\ell] \leq \mathbb{E}[P_1]$. The following then completes the proof of the lemma:

$$\mathbb{E}[c(S)] = \mathbb{E}[c(S_\ell) + \alpha\text{OPT}(f_\ell)] = \mathbb{E}[P_\ell] \leq \mathbb{E}[P_1] = c(S_1) + \alpha\text{OPT}(f_1) \leq \alpha\text{OPT}(f).$$

Here, the final inequality follows by observing that $f_1 = \text{DENSEDECOMPOSITIONPREPROCESS}(f, c_f(1 + \epsilon)\rho)$ and applying Lemma 19(2). \blacktriangleleft

4 SubmodCover and SupmodDD

In this section, we prove Theorems 5 and 8.

► **Theorem 5.** *Let $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ be an integer-valued normalized supermodular function and ρ be a rational number. Then, there exists a normalized non-decreasing submodular function $h : 2^V \rightarrow \mathbb{R}_{\geq 0}$ such that for $F \subseteq V$, we have that $\lambda_{f|_{V-F}}^* \leq \rho$ if and only if $h(F) \geq h(V)$. Moreover,*

1. *if ρ is an integer, then h is integer-valued,*
2. *$h(v) \leq \max\{0, f(v|V-v) - \rho\}$ for all $v \in V$, and*
3. *evaluation queries for the function h can be answered in polynomial time by making polynomial number of evaluation queries to the function f .*

Proof. For simplicity, we define an intermediate function $g : 2^V \rightarrow \mathbb{R}_{\geq 0}$, and use it to define the function $h : 2^V \rightarrow \mathbb{R}_{\geq 0}$ of interest. The functions g and h are as follows: for every $X \subseteq V$,

$$\begin{aligned} g(X) &:= \max\{f(Z) - \rho|Z| : Z \subseteq X\}, \text{ and} \\ h(X) &:= g(V) - g(V - X). \end{aligned}$$

The following claim shows that the function h is normalized, non-decreasing, and submodular.

▷ **Claim 27.** The function h is normalized, non-decreasing, and submodular.

Proof. We note that h is normalized by definition, and is non-decreasing because the g is non-decreasing. To show that h is submodular, it suffices to show that the g is supermodular. Let $A, B \subseteq V$ be arbitrary. Let $Z_A := \arg \max\{f(Z) - \rho|Z| : Z \subseteq A\}$ and $Z_B := \arg \max\{f(Z) - \rho|Z| : Z \subseteq B\}$. Then, we have the following:

$$\begin{aligned} g(A) + g(B) &= f(Z_A) - \rho|Z_A| + f(Z_B) - \rho|Z_B| \\ &\leq f(Z_A \cup Z_B) + f(Z_A \cap Z_B) - \rho|Z_A \cup Z_B| - \rho|Z_A \cap Z_B| \\ &= (f(Z_A \cup Z_B) - \rho|Z_A \cup Z_B|) + (f(Z_A \cap Z_B) - \rho|Z_A \cap Z_B|) \\ &\leq g(A \cup B) + g(A \cap B). \end{aligned}$$

Here, the first inequality follows from supermodularity of f and the second inequality is by definition of g . \triangleleft

Let $F \subseteq V$. We now show that $\lambda_{F|V-F}^* \leq \rho$ if and only if $h(F) \geq h(V)$ as claimed. We have the following sequence of equivalences.

$$\begin{aligned} \lambda_{F|V-F}^* \leq \rho &\Leftrightarrow \max_{S \subseteq V-F} \left\{ \frac{f(S)}{|S|} \right\} \leq \rho \\ &\Leftrightarrow g(V-F) \leq 0 \\ &\Leftrightarrow g(V) - g(V-F) - g(V) + g(\emptyset) \geq 0 \\ &\Leftrightarrow h(F) - h(V) \geq 0. \end{aligned} \tag{2}$$

Here, the second equivalence can be seen by the following. For the forward direction, we suppose that $\max \left\{ \frac{f|_{V-F}(S)}{|S|} : S \subseteq V-F \right\} \leq \rho$. By way of contradiction, suppose that $g(V-F) > 0$. By definition of the function g , there exists a set $Z^* \subseteq V-F$ such that $g(V-F) = f(Z^*) - \rho|Z^*|$. Thus, $f(Z^*) - \rho|Z^*| > 0$. Equivalently, we have that $f(Z^*)/|Z^*| > \rho$, a contradiction to our hypothesis. Here we note that $Z^* \neq \emptyset$ as otherwise we would have that $f(Z^*) - \rho|Z^*| = 0$ since our function f is normalized, contradicting our choice of Z^* . For the reverse direction, suppose that $g(V-F) \leq 0$. By way of contradiction, suppose that there exists a non-empty set $S^* \subseteq V-F$ such that $f(S^*)/|S^*| > \rho$. Then, we equivalently have that $f(S^*) - \rho|S^*| > 0$, a contradiction.

We now show that the function h satisfies properties (1)-(3) of the theorem. We note that if ρ is an integer, then h is integer-valued by definition. Moreover, we can answer evaluation queries for h using polynomial many evaluation queries to f (via supermodular maximization). Thus h satisfies properties (1) and (3). We now show property (2).

Let $v \in V$. We have the following:

$$\begin{aligned} h(v) &= g(V) - g(V-v) \\ &= \max \{f(Z) - \rho|Z| : Z \subseteq V\} - \max \{f(Z') - \rho|Z'| : Z' \subseteq V-v\} \\ &= \max \left\{ 0, \max \{f(Z) - \rho|Z| : v \in Z \subseteq V\} - \max \{f(Z') - \rho|Z'| : Z' \subseteq V-v\} \right\} \\ &\leq \max \left\{ 0, \max \{(f(Z) - \rho|Z|) - (f(Z') - \rho|Z'|) : v \in Z \subseteq V \text{ and } Z' \subseteq V-v\} \right\} \\ &\leq \max \left\{ 0, \max \{(f(Z) - \rho|Z|) - (f(Z-v) - \rho|Z-v|) : v \in Z \subseteq V\} \right\} \\ &= \max \left\{ 0, \max \{(f(Z) - f(Z-v)) : v \in Z \subseteq V\} - \rho \right\} \\ &\leq \max \{0, f(V) - f(V-v) - \rho\}, \end{aligned}$$

where the final inequality is because $f(V-v) + f(Z) \leq f(V) + f(Z-v)$ for all $Z \subseteq V$ such that $v \in Z$ by supermodularity of the function f . ◀

► **Theorem 8.** *Let $h : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ be an integer-valued normalized non-decreasing submodular function. Then, there exists a normalized supermodular function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ such that for $F \subseteq V$, we have that $h(F) \geq h(V)$ if and only if $\lambda_{F|V-F}^* \leq 1$. Moreover,*

1. $f(v|V-v) = h(v) + 1$ for all $v \in V$, and
2. evaluation queries for the function f can be answered in polynomial time by making a constant number of evaluation queries to the function h .

Proof. We consider the function $f : 2^V \rightarrow \mathbb{Z}_{\geq 0}$ defined as follows: for every $X \subseteq V$,

$$f(X) := h(V) - h(V-X) + |X|.$$

We note that the function f is normalized, non-decreasing, integer-valued and supermodular. Moreover, we can answer evaluation queries for the function f using two queries to the

evaluation oracle for h . We note that for $F \subseteq V$, $h(F) \geq h(V)$ if and only if $\lambda_{f|_{V-F}}^* \leq 1$ can be observed by following the steps of (2) in reverse order, and so we omit the formal details here for brevity. Property (1) of the theorem can be observed as follows: for $v \in V$, we have the following:

$$\begin{aligned} f(v|V-v) &= f(V) - f(V-v) = (h(V) - h(\emptyset) + |V|) - (h(V) - h(v) + |V-v|) \\ &= h(v) + 1, \end{aligned}$$

where the final equality is because h is normalized. ◀

5 Conclusion

We considered several interrelated density deletion problems motivated by the question of understanding the robustness of densest subgraph. We showed tight logarithmic approximations for these problems. We showed inapproximability of graph density deletion by reduction from set cover, and approximation algorithms by exhibiting the equivalence of supermodular density deletion and submodular cover. Motivated by the hardness results, we designed bicriteria approximations. Our bicriteria approximation for graph density deletion is LP-based, and that for supermodular density deletion is a randomized combinatorial one and relies on the notion of dense decomposition of supermodular functions. We mention two open questions. First, our bicriteria approximation for supermodular density deletion depends on the parameter c_f related to the input supermodular function (see Theorem 11). Is it possible to design a bicriteria approximation without dependence on the parameter c_f ? Second, our hardness reduction shows that ρ -GRAPHDD is $\Omega(\log n)$ -hard for every fixed constant integer $\rho \geq 2$. We were able to adapt our reduction to conclude that it is $\Omega(\log n)$ -hard for every fixed constant $\rho \geq 3$ (not necessarily integers). Is ρ -GRAPHDD $\Omega(\log n)$ -hard for every fixed constant $\rho > 1$?

References

- 1 Vineet Bafna, Piotr Berman, and Toshihiro Fujito. Constant ratio approximations of the weighted feedback vertex set problem for undirected graphs. In *Algorithms and Computations*, pages 142–151, 1995. doi:10.1007/BFB0015417.
- 2 Cristina Bazgan, André Nichterlein, and Sofia Vazquez Alferez. Destroying Densest Subgraphs Is Hard. In *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, pages 6:1–6:17, 2024. doi:10.4230/LIPICS.SWAT.2024.6.
- 3 Ann Becker and Dan Geiger. Optimization of Pearl’s method of conditioning and greedy-like approximation algorithms for the vertex feedback set problem. *Artificial Intelligence*, 83:167–188, 1996. doi:10.1016/0004-3702(95)00004-6.
- 4 Digvijay Boob, Yu Gao, Richard Peng, Saurabh Sawlani, Charalampos Tsourakakis, Di Wang, and Junxing Wang. Flowless: Extracting densest subgraphs without flow computations. In *Proceedings of The Web Conference 2020*, pages 573–583, 2020. doi:10.1145/3366423.3380140.
- 5 Karthekeyan Chandrasekaran, Chandra Chekuri, Samuel Fiorini, Shubhang Kulkarni, and Stefan Weltge. Polyhedral aspects of feedback vertex set and pseudoforest deletion set. *Mathematical Programming*, pages 1–48, 2025.
- 6 Karthekeyan Chandrasekaran, Chandra Chekuri, and Shubhang Kulkarni. On deleting vertices to reduce density in graphs and supermodular functions, 2025. doi:10.48550/arXiv.2503.08828.

- 7 Moses Charikar. Greedy Approximation Algorithms for Finding Dense Components in a Graph. In *Approximation Algorithms for Combinatorial Optimization*, pages 84–95, 2000. doi:10.1007/3-540-44436-X_10.
- 8 Chandra Chekuri, Aleksander Bjørn Christiansen, Jacob Holm, Ivor van der Hoog, Kent Quanrud, Eva Rotenberg, and Chris Schwiegelshohn. Adaptive out-orientations with applications. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3062–3088, 2024.
- 9 Chandra Chekuri, Kent Quanrud, and Manuel R Torres. Densest subgraph: Supermodularity, iterative peeling, and flow. In *Proceedings of the 2022 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 1531–1555, 2022. doi:10.1137/1.9781611977073.64.
- 10 Fabián A. Chudak, Michel X. Goemans, Dorit S. Hochbaum, and David P. Williamson. A primal–dual interpretation of two 2-approximation algorithms for the feedback vertex set problem in undirected graphs. *Operations Research Letters*, 22(4):111–118, 1998. doi:10.1016/S0167-6377(98)00021-2.
- 11 William H Cunningham. Optimal attack and reinforcement of a network. *Journal of the ACM (JACM)*, 32(3):549–561, 1985. doi:10.1145/3828.3829.
- 12 Laxman Dhulipala, Quanquan C Liu, Sofya Raskhodnikova, Jessica Shi, Julian Shun, and Shangdi Yu. Differential privacy from locally adjustable graph algorithms: k-core decomposition, low out-degree ordering, and densest subgraphs. In *2022 IEEE 63rd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 754–765, 2022. doi:10.1109/FOCS54457.2022.00077.
- 13 Michael Dinitz, Satyen Kale, Silvio Lattanzi, and Sergei Vassilvitskii. Almost tight bounds for differentially private densest subgraph. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 2908–2950, 2025. doi:10.1137/1.9781611978322.94.
- 14 Irit Dinur and David Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing, STOC*, pages 624–633, 2014. doi:10.1145/2591796.2591884.
- 15 Uriel Feige. A threshold of $\ln n$ for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998. doi:10.1145/285055.285059.
- 16 A. Frank. *Connections in combinatorial optimization*. Oxford University Press, Oxford, 2011.
- 17 Satoru Fujishige. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980. doi:10.1287/MOOR.5.2.186.
- 18 Toshihiro Fujito. Approximating node-deletion problems for matroidal properties. *Journal of Algorithms*, 31(1):211–227, 1999. doi:10.1006/JAGM.1998.0995.
- 19 A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, USA, 1984.
- 20 Elfarouk Harb, Kent Quanrud, and Chandra Chekuri. Faster and scalable algorithms for densest subgraph and decomposition. In *Advances in Neural Information Processing Systems*, 2022.
- 21 Elfarouk Harb, Kent Quanrud, and Chandra Chekuri. Convergence to lexicographically optimal base in a (contra)polymatroid and applications to densest subgraph and tree packing. In *31st Annual European Symposium on Algorithms (ESA)*, pages 56:1–56:17, 2023. doi:10.4230/LIPICSESA.2023.56.
- 22 Subhash Khot and Oded Regev. Vertex cover might be hard to approximate to within $2 - \epsilon$. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.
- 23 Tommaso Lanciano, Atsushi Miyachi, Adriano Fazzino, and Francesco Bonchi. A survey on the densest subgraph problem and its variants. *ACM Comput. Surv.*, 56(8), April 2024. doi:10.1145/3653298.
- 24 Audrey Lee and Ileana Streinu. Pebble game algorithms and sparse graphs. *Discrete Mathematics*, 308(8):1425–1437, April 2008. doi:10.1016/J.DISC.2007.07.104.

- 25 John M. Lewis and Mihalis Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, April 1980. doi:10.1016/0022-0000(80)90060-4.
- 26 Ta Duy Nguyen and Alina Ene. Multiplicative weights update, area convexity and random coordinate descent for densest subgraph problems. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.
- 27 Jean-Claude Picard and Maurice Queyranne. A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory. *Networks*, 12(2):141–159, 1982. doi:10.1002/NET.3230120206.
- 28 Saurabh Sawlani and Junxing Wang. Near-optimal fully dynamic densest subgraph. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 181–193, 2020. doi:10.1145/3357713.3384327.
- 29 Shuichi Ueno, Yoji Kajitani, and Shin'ya Gotoh. On the nonseparating independent set problem and feedback set problem for graphs with no vertex degree exceeding three. *Discrete Mathematics*, 72(1):355–360, December 1988. doi:10.1016/0012-365X(88)90226-9.
- 30 Nate Veldt, Austin R Benson, and Jon Kleinberg. The generalized mean densest subgraph problem. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1604–1614, 2021. doi:10.1145/3447548.3467398.
- 31 Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982. doi:10.1007/BF02579435.
- 32 Michał Włodarczyk. Losing treewidth in the presence of weights. In *Proceedings of the 2025 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3743–3761, 2025.