On Deleting Vertices to Reduce Density in Graphs and Supermodular Functions*

Karthekeyan Chandrasekaran Chandra Chekuri Shubhang Kulkarni

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 $f : 2^V \to \mathbb{Z}_{\geq 0}$ via an evaluation oracle with element costs and a non-negative integer $\rho \geq 0$ and the goal is 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.

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. 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, 7, 8, 11, 12, 19, 22, 25, 27, 29]. A useful feature of DSG is its polynomial-time solvability. This was first seen via a reduction to network flow [18, 26] but another way to see it is by considering a more general problem, namely the *densest supermodular subset* problem (DSS): Given a supermodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ 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 $f : 2^V \to \mathbb{Z}$ 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 [8, 19, 20, 29].

*Grainger College of Engineering, University of Illinois, Urbana-Champaign, Urbana, IL 61801, Email: {karthe, chekuri, smkulka2}@illinois.edu. Supported in part by NSF grant CCF-2402667.

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 1.1 (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 $c : V \to \mathbb{R}_{\geq 0}$*
Goal:	$\arg\min\{\sum_{u \in S} c_S : S \subseteq V$ *and* $\lambda^*_{G-S} \leq \rho\}$.

This deletion problem naturally generalizes to supermodular functions as defined below. We recall that a set function $f : 2^V \to \mathbb{R}$ 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 $f : 2^V \to \mathbb{R}$ 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 1.2 (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 $f : 2^V \to \mathbb{Z}_{\geq 0}$ via evaluation oracle and element costs $c : V \to \mathbb{R}_{\geq 0}$*
Goal:	$\arg\min\{\sum_{u \in S} c_u : S \subseteq V$ *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 [24]. 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 Alferez [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 [10] 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. Vertex cover, PFDS, and FVS admit 2-approximations, and moreover this bound cannot be improved under the Unique Games Conjecture (UGC) [21]. We note that while 2-approximations for vertex cover are relatively easy, 2-approximations for FVS and PFDS are non-obvious [1,3,9]. 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 [6]. Fujito [17] 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 MatroidFVS[1], is defined below.

Definition 1.3 (MatroidFVS). *The Matroid Feedback Vertex Set problem, denoted MatroidFVS, is defined as follows:*

Input:	*Graph $G = (V, E)$, vertex costs $c : V \to \mathbb{R}_{\geq 0}$, 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 [17] obtained a 2-approximation for MatroidFVS for the class of "uniformly sparse" matroids [23]. 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 $\mathcal{M}_\rho$ is the ρ-fold union of the 1-cycle matroid defined on the edge set of the input graph (see Theorem A.2 in Appendix A). Although it is not obvious, we can show that MatroidFVS is a special case of 1-SupmodDD (see Theorem A.1 in Appendix A). We refer the reader to 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 1.4 (SubmodCover). *The submodular cover problem, denoted SubmodCover, is defined as follows:*

Input:	*Integer-valued normalized non-decreasing submodular function $h : 2^V \to \mathbb{Z}_{\geq 0}$* *via evaluation oracle and* *element costs $c : V \to \mathbb{R}_{\geq 0}$*
Goal:	$\arg\min\{\sum_{e \in F} c_e : F \subseteq V \text{ and } h(F) \geq h(V)\}$.

For a function $f : 2^V \to \mathbb{R}$, 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 1.1. *Let $f : 2^V \to \mathbb{Z}_{\geq 0}$ be an integer-valued normalized supermodular function and ρ be a rational number. Then, there exists a normalized non-decreasing submodular function $h : 2^V \to \mathbb{R}_{\geq 0}$ such that*

[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.

1. *if ρ is an integer, then h is integer-valued,*

2. *for $F \subseteq V$, we have that $\lambda^*_{f|_{V-F}} \le \rho$ if and only if $h(F) \ge h(V)$,*

3. *$h(v) \le \max\{0, f(v|V-v) - \rho\}$ for all $v \in V$, and*

4. *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 1.1 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 1.1, 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 1.1. *ρ-SupmodDD for integer-valued ρ admits an $(1 + \ln(\max_{v\in V} f(v|V-v)))$-approximation, where $f : V \to \mathbb{Z}_{\ge 0}$ 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 1.1. *The reduction from SupmodDD to SubmodCover is in some sense implicit in prior literature (see [17, 28] 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 1.2. *Let $h : 2^V \to \mathbb{Z}_{\ge 0}$ be an integer-valued normalized non-decreasing submodular function. Then, there exists a normalized supermodular function $f : 2^V \to \mathbb{Z}_{\ge 0}$ such that*

1. *for $F \subseteq V$, we have that $h(F) \ge h(V)$ if and only if $\lambda^*_{f|_{V-F}} \le 1$,*

2. *$f(v|V-v) = h(v) + 1$ for all $v \in V$, and*

3. *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 [17]. 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 Theorem A.2). 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 $\rho \ge 2$, ρ-GraphDD is $\Omega(\log n)$-hard to approximate via a reduction from Set Cover.

Theorem 1.3. *For every integer $\rho \ge 2$, there is no $o(\log n)$ approximation for ρ-GraphDD assuming $P \ne NP$, 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 [17]) 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 1.1. 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.

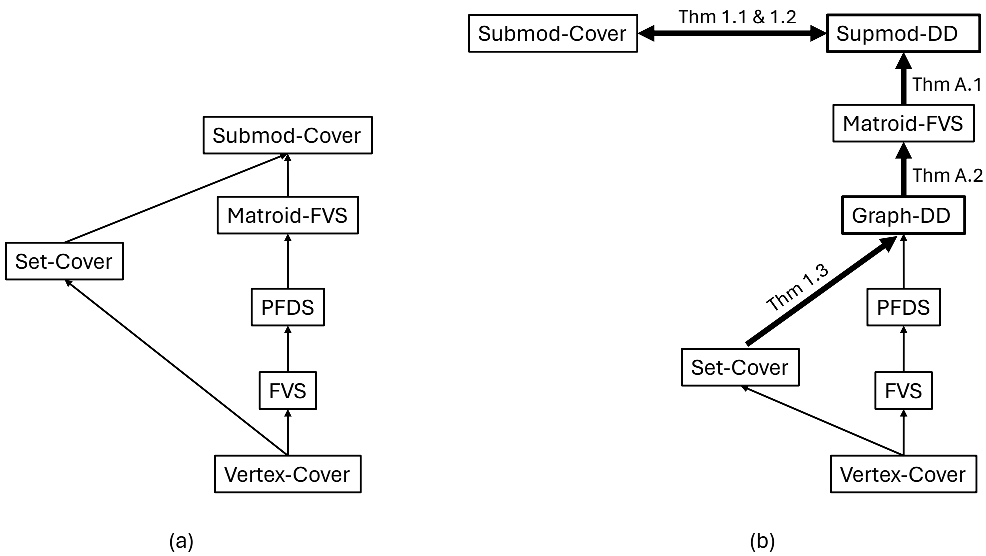

Figure 1: Reductions between problems of interest to this work. Arrow from Problem A to Problem B implies that Problem A has an approximation-preserving reduction to Problem B. Figure (a) consists of the connections between problems known prior to our work. Figure (b) showcases our results.

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.

Theorem 1.4. *There exists a polynomial time algorithm which takes as input a graph $G = (V, E)$, vertex deletion costs $c : V \to \mathbb{R}_{\geq 0}$, 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. $\sum_{u \in S} c_u \leq \left(\frac{1}{\epsilon}\right) \cdot \texttt{OPT}$,

where $\texttt{OPT}$ 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 $f : 2^V \to \mathbb{R}_{\geq 0}$, 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 [8]. 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 1.5. *There exists a randomized polynomial time algorithm which takes as input a normalized monotone supermodular function $f : 2^V \to \mathbb{Z}$ (given by oracle access), element deletion costs $c : V \to \mathbb{R}_{\geq 0}$, 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. $\mathbb{E}\left[\sum_{u \in S} c_u\right] \leq c_f \left(1 + \frac{1}{\epsilon}\right) \cdot \mathtt{OPT}$,

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

As a consequence of Theorem 1.5, 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 1.4. We discuss another special case of SupmodDD where the supermodular function of interest has bounded c_f to illustrate the significance of Theorem 1.5. 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 [29]. Subsequent work by Chekuri, Quanrud, and Torres [8] showed that p-mean density is a special case of the densest supermodular set problem (i.e., DSS) where the supermodular function $f : 2^V \to \mathbb{R}_{\geq 0}$ 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 1.5 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. The main body of the paper is organized as follows. In Section 1.2, we give preliminaries which will be used throughout the technical sections. In Section 2, we show an approximation-preserving reduction from SetCover to ρ-GraphDD and prove Theorem 1.3. In Section 3, we give bicriteria approximations for ρ-GraphDD and ρ-SupmodDD. Here, we prove Theorem 1.4 in Section 3.1 and Theorem 1.5 in Section 3.2. Finally, in Section 4 we show the connections between SupmodDD and Submodular Cover by proving Theorems 1.1 and 1.2.

1.2 Preliminaries: Characterizing Density Using Orientations

In this section, we give a characterization of density via *(fractional) orientations* which we use throughout the technical sections. We recall that an *orientation* of a graph $G = (V, E)$ assigns each edge $\{u, v\}$ to either

u or v. A *fractional orientation* is an assignment of two non-negative numbers $z_{e,u}$ and $z_{e,v}$ for each edge $e = \{u, v\} \in E$ such that $z_{e,u} + z_{e,v} = 1$. We note that an orientation is a fractional orientation where all values in the assignment are either 0 or 1. For notational convenience, we use $\vec{G} = (V, \vec{E})$ to denote an orientation of the graph G and $d^{\text{in}}_{\vec{G}}(u)$ to denote the indegree of a vertex $u \in V$ in the oriented graph $\vec{G}$.

The following connection between density and (fractional) orientations will be used in our hardness reduction in Section 2, our ILP in Section 3.1, and the connection between ρ-GraphDD and MatroidFVS in Appendix A.

Proposition 1.1. *Let $G = (V, E)$ be a graph. Then, we have the following:*

1. *let $\rho \in \mathbb{R}_{\geq 0}$ be a real value; then, $\lambda^*_G \leq \rho$ if and only if there exists a fractional orientation z such that $\sum_{e \in \delta(u)} z_{e,u} \leq \rho$ for every $u \in V$, and*
2. *let $\rho \in \mathbb{Z}_{\geq 0}$ be an integer value; then, $\lambda^*_G \leq \rho$ if and only if there exists an orientation $\vec{G} = (V, \vec{E})$ of the graph G such that $d^{in}_{\vec{G}}(u) \leq \rho$ for every $u \in V$.*

The first part of the proposition states that a graph G has density at most ρ if and only if the edges of G can be fractionally oriented such that the total fractional in-degree at every vertex is at most ρ. This characterization is implied by the dual of Charikar's LP [6] to solve DSG. The second part of the proposition is a strengthening of the first part when ρ is an integer and states that a graph G has density at most ρ iff the edges of G can be (integrally) oriented such that the in-degree at every vertex is at most ρ. This characterization is implied by a result of Hakimi [15] and also via the dual of Charikar's LP [6].

2 Approximation Hardness

In this section, we show Theorem 1.3, i.e., we show an approximation preserving reduction from SetCover to ρ-GraphDD. We recall the set cover problem and its inapproximability.

Definition 2.1 (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 $c : \mathcal{S} \to \mathbb{R}_{\geq 0}$*
Goal:	$\arg\min\{\sum_{S \in \mathcal{F}} c_e : \mathcal{F} \subseteq \mathcal{S}$ *with* $\cup_{S \in \mathcal{F}} S = \mathcal{U}\}$.

Theorem 2.1. *[13, 14] 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.*

Warmup. As a warmup, we describe the idea underlying the reduction to prove Theorem 1.3 by giving the proof of a weaker hardness result—we will show that there is no $(\rho + 1 - \epsilon)$-approximation for ρ-GraphDD unless $P = NP$. In particular, given a set cover instance $(\mathcal{S}, \mathcal{U}, c)$, we will construct a graph $H = (V_H, E_H)$ with vertex deletion costs $c_H : V \to \mathbb{R}$ such that a $(f_{\max} - 1)$-GraphDD of H corresponds exactly to a set cover of $(\mathcal{S}, \mathcal{U})$, where $f_{\max}$ is the maximum frequency of an element. Since there is no $(f_{\max} - \epsilon)$-approximation for SetCover, we obtain the claimed result. For simplicity (and without loss of generality), we henceforth assume that every element has the same frequency $f_{\max}$. Our reduction proceeds by considering the incidence graph of the set cover instance—this is the bipartite graph where there is a vertex v_S for each set $S \in \mathcal{S}$, vertex u_e for each element $e \in \mathcal{U}$, and edges (v_S, u_e) for each $S \in \mathcal{S}$ and $e \in \mathcal{U}$ such that $e \in S$. We make a simple modification to this incidence graph to obtain the graph $H = (V_H, E_H)$: for every set $S \in \mathcal{S}$, we add $f_{\max} - 1$ self loops to the vertex v_S. We also define the the cost function as $c_H(v_S) = c(S)$ for all $S \in \mathcal{S}$ and $c(u_e) = \infty$ for all $e \in \mathcal{U}$.

Remark 2.1. *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.*

The intuition behind the addition of self-loops is that the subgraph induced by every element-vertex and its neighborhood (which contains exactly $f_{\max}$ set-vertices) is strictly larger than ρ. This is because the number of edges here is $f_{\max}(1 + f_{\max})$ whereas the number of vertices is $(1 + f_{\max})$. Consequently, any $(f_{\max} - 1)$-GraphDD must delete at least one vertex from this induced subgraph. Our cost function ensures that the deleted vertices are set-vertices, and so these correspond to a set cover. We argue the reverse direction of the reduction by leveraging the connection between integer density and integral orientations. 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 $H - X_F$ has density at most ρ by exhibiting a simple orientation in which the indegree of every vertex is at most ρ: for every remaining edge of type (v_S, u_e), orient the edge from v_S to u_e. Orient all self-loops arbitrarily. We note that the indegree of every vertex v_S is at most ρ since there are only ρ self loops by construction. By way of contradiction, suppose that the indegree of a vertex u_e was strictly larger than ρ. We note that by construction, $d_H(u_e) = \rho + 1$. Consequently, no neighbor of u_e is in X_F. Thus, F is not a set cover, a contradiction.

Our proof of Theorem 1.3 follows the same high-level strategy: obtain an appropriate modification of the incidence graph so that for every element, there is an appropriate subgraph with density larger than the target density. However, in contrast to the situation in the reduction we described above, Theorem 1.3 is a statement about all *fixed* integer target densities at least 2 and so we cannot set the target density as a parameter in our reduction. This leads to additional complications which we overcome by replacing the vertices and edges of the incidence graph with appropriate (binary tree) gadgets. We now restate and prove Theorem 1.3.

Theorem 1.3. *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. TOPROVE 0 ◻

3 Bicriteria Approximations

In this section, we design bicriteria approximation algorithms for GraphDD and SupmodDD.

3.1 Bicriteria for GraphDD

In this section, we extend the ideas of [5] to obtain an ILP formulation for GraphDD. We then give a simple threshold-rounding algorithm for its LP relaxation to prove Theorem 1.4.

3.1.1 Orientation ILP

Our ILP for ρ-GraphDD is based on the characterization of λ^*_G via fractional orientations given in Proposition 1.1. We recall that fractional orientation is an assignment of two non-negative numbers $z_{e,u}$ and $z_{e,v}$ for each edge $\{u, v\} \in E$ such that $z_{e,u} + z_{e,v} = 1$, and Proposition 1.1(1) states that a graph G has density at most ρ if and only if the edges of G can be fractionally oriented such that the total fractional in-degree at every vertex is at most ρ.

We describe the details of our formulation now. For an edge $e = uv$ we use variables $z_{e,u}$ and $z_{e,v}$ to denote the fractional amount of e that is oriented towards u and v respectively. Since the ρ-GraphDD is a vertex deletion problem, we also have variables x_u for each $u \in V$ to indicate whether u is deleted. An edge $e = uv$ is in the residual graph only if u and v are not deleted. These observations allow us to formulate the ILP for ρ-GraphDD below.

$$
\begin{array}{lll}
\min & \sum_{u\in V} c_u x_u & \\
\text{s.t.} & x_u + x_v + z_{e,u} + z_{e,v} \geq 1 & \forall e = uv \in E \\
& \rho x_u + \sum_{e\in\delta(u)} z_{e,u} \leq \rho & \forall u \in V \\
& z_{e,u} \geq 0 & \forall u \in V, \forall e \in \delta(u) \\
& x_u \in \{0,1\} & \forall u \in V
\end{array}
$$

We will denote the LP-relaxation of the above ILP for the instance (G, c, ρ) as $\text{LP}_{\text{orient}}(G, c, \rho)$.

3.1.2 Rounding the Orientation LP

We give our LP-rounding based bicriteria algorithm for ρ-GraphDD and prove Theorem 1.4. In particular, we consider the simple threshold-rounding based algorithm for the orientation LP given in Algorithm 1. Let S denote the set returned by Algorithm 1. Lemma 3.1 shows that the cost of the set S is at most $1/\epsilon$ times the cost of an optimum solution and thus satisfies property (2) of the theorem. Lemma 3.2 shows that the density of the graph $G - S$ is at most $\rho/(1-2\epsilon)$ and so the set S satisfies property (1) of the theorem. Algorithm 1, Lemma 3.1 and Lemma 3.2 together complete the proof of Theorem 1.4.

Algorithm 1 Bicriteria approximation algorithm for GraphDD

Input: (1) Graph $G = (V, E)$, (2) Costs $c : V \to \mathbb{R}_+$, (3) Target $\rho \in \mathbb{Z}_+$, (4) Error parameter $\epsilon \in (0, 1/2)$

1. Let (x, z) be an optimal solution to $\text{LP}_{\text{orient}}(G, c, \rho)$.
2. **return** $S := \{u \in V : x_u > \epsilon\}$.

Lemma 3.1 (Approximate Cost)**.** $c(S) \leq \frac{1}{\epsilon}\sum_{u\in V} c_u x_u$.

Proof. TOPROVE 1 □

Lemma 3.2 (Approximate Feasibility)**.** $\lambda^*_{G-S} \leq \rho \cdot \frac{1}{1-2\epsilon}$.

Proof. TOPROVE 2 □

Claim 3.1. $z'_{e,u} + z'_{e,v} \geq 1$ *for all* $e = uv \in E[V - S]$.

Proof. TOPROVE 3 □

Claim 3.2. $\sum_{e\in\delta_{V-S}(u)} z'_{e,u} \leq \rho \cdot \frac{1}{1-2\epsilon}$.

Proof. TOPROVE 4 □

3.2 Bicriteria for SupmodDD

In this section, we describe a randomized combinatorial bicriteria approximation algorithm for ρ-SupmodDD and prove Theorem 1.5. 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* $\pi : V \to \mathbb{R}_{\geq 0}$ 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 1.5. 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 3.2.1 we give our preprocessing step based on the dense decomposition of supermodular functions. In Section 3.2.2 we describe our element potentials based on marginal gains of supermodular functions. In Section 3.2.3 we present our algorithm and complete the proof of Theorem 1.5.

3.2.1 Preprocessing via Dense Decomposition

We discuss the *dense decomposition* of a normalized non-negative supermodular set function [16, 19] and prove a lemma that will enable us to use it as a preprocessing step.

Definition 3.1. *[19] Let $f : 2^V \to \mathbb{R}_+$ be a non-negative normalized supermodular function. A sequence $(V_1, \rho_1), (V_2, \rho_2), \ldots, (V_k, \rho_k)$ is the* dense decomposition of f *if*

1. *$V_1, \ldots, V_k$ is a partition of V obtained iteratively as follows: for $i = 1, 2, \ldots, k$, V_i is the inclusion-wise maximal set $S \subseteq V - \cup_{j\in[i-1]} V_j$ 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 $\rho_1, \ldots, \rho_k$ are obtained as*
$$\rho_i := \frac{f\left(\bigcup_{j\in[i]} V_j\right) - f\left(\bigcup_{j\in[i-1]} V_j\right)}{|V_i|} \ \forall\, i \in [k].$$

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 $f_{/S} : 2^{V-S} \to \mathbb{R}$ 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 3.3. *Let $f : 2^V \to \mathbb{R}_{\geq 0}$ be a normalized non-negative supermodular function, $c : V \to \mathbb{R}_+$ be a cost function, and $\rho \in \mathbb{R}_+$ be a positive real value. Moreover, let $(V_1, \phi_1), (V_2, \phi_2), \ldots, (V_k, \phi_k)$ 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. *$\texttt{OPT}(f) \geq \texttt{OPT}(f|_R)$, where $\texttt{OPT}(f)$ and $\texttt{OPT}(f|_R)$ 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. TOPROVE 5 ∎

3.2.2 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 3.4. *Let $\rho \in \mathbb{R}$ and $\epsilon \in (0,1)$. Let $f : 2^V \to \mathbb{Z}$ 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. TOPROVE 6 □

3.2.3 Random Deletion Algorithm

We now describe our bicriteria algorithm for ρ-SupmodDD and analyze its approximation factor. Algorithm 2, Lemma 3.5 and Lemma 3.6 together complete the proof of Theorem 1.5.

Algorithm. Our algorithm takes as input (1) a normalized non-negative supermodular function $f : 2^V \to \mathbb{R}_+$, (2) element deletion costs $c : V \to \mathbb{R}_+$, (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 $(V_1,\phi_1),(V_2,\phi_2),\ldots,(V_k,\phi_k)$ 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 $f|_R : 2^R \to \mathbb{R}_{\geq 0}$—we use DenseDecompositionPreprocess(f,ρ) to denote a subroutine that computes the set R and returns the tuple $(f|_R, R)$. Next, the algorithms 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 2.

Algorithm 2 Bicriteria approximation algorithm for ρ-SupmodDD

Algorithm$\Big(\left(f : 2^V \to \mathbb{R}, c\right), \rho, \epsilon\Big)$:

1. $S := \emptyset$
2. **while** $\lambda_f^* > c_f(1+\epsilon)\rho$:
   (a) Redefine $(f,V) :=$ DenseDecompositionPreprocess$(f, c_f(1+\epsilon)\rho)$
   (b) $u :=$ vertex sampled from V according to the following distribution:
   $$\Pr(u=v) := \frac{f(v|V-v)}{c(v)\cdot W}\ \forall v \in V, \text{ where } W := \sum_{v\in V}\frac{f(v|V-v)}{c(v)} \text{ is a normalizing factor}$$
   (c) $S := S+u$ and $f := f|_{V-u}$
3. **return** S.

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

Definition 3.2. 1. *A sequence of random variables $P_1, P_2, \ldots$ is called a* supermartingale *w.r.t. the sequence $X_1, X_2, \ldots$ of random variables if for each $i \in \mathbb{Z}_+$ it holds that (i) P_i is a function of $X_1, \ldots X_i$, (ii) $\mathbb{E}[|P_i|] < \infty$ and (iii) $\mathbb{E}[P_{i+1}|X_1, \ldots X_i] \leq P_i$.*

2. *A random variable T is called a* stopping time *with respect to the sequence of random variables $P_1, P_2, \ldots$ if for each $i \in \mathbb{Z}_+$, the event $(T \leq i)$ depends only on $P_1, \ldots, P_i$.*

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 2.

Theorem 3.1 (Doob's Optional-Stopping Theorem)**.** *Let $P_0, P_1, \ldots$ be a supermartingale w.r.t. the sequence $X_1, X_2, \ldots$ 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 2 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)), $f_i : 2^{V_i} \to \mathbb{R}_{\geq 0}$ 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 3.3(1) while inducting on ℓ. We omit details of the proof here for brevity.

Lemma 3.5 (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 $\texttt{OPT}(g)$ to denote the value of an optimal ρ-SupmodDD for g with respect to the cost function c.

Lemma 3.6 (Approximate Cost)**.** $\mathbb{E}[c(S)] \leq c_f(1+1/\epsilon)\texttt{OPT}(f)$.

Proof. TOPROVE 7 □

4 SubmodCover and SupmodDD

In this section, we prove Theorems 1.1 and 1.2.

Theorem 1.1. *Let $f : 2^V \to \mathbb{Z}_{\geq 0}$ be an integer-valued normalized supermodular function and ρ be a rational number. Then, there exists a normalized non-decreasing submodular function $h : 2^V \to \mathbb{R}_{\geq 0}$ such that*

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

Proof. TOPROVE 8 □

Theorem 1.2. *Let $h : 2^V \to \mathbb{Z}_{\geq 0}$ be an integer-valued normalized non-decreasing submodular function. Then, there exists a normalized supermodular function $f : 2^V \to \mathbb{Z}_{\geq 0}$ such that*

1. *for $F \subseteq V$, we have that $h(F) \geq h(V)$ if and only if $\lambda^*_{f|_{V-F}} \leq 1$,*
2. *$f(v|V - v) = h(v) + 1$ for all $v \in V$, and*
3. *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. TOPROVE 9 □

5 Conclusion

In this work, 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 our hardness results, we designed bicriteria approximation. Our bicriteria approximation for graph density deletion is LP-based and that for supermodular density deletion is randomized, combinatorial, and relies on the notion of dense decomposition of supermodular functions. We mention two open questions raised by our work. Firstly, we note that our bicriteria approximation for supermodular density deletion depends on the parameter c_f related to the input supermodular function (see Theorem 1.5). Is it possible to design a bicriteria approximation without the dependence on the parameter c_f? Secondly, we note that 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 it $\Omega(\log n)$-hard for every fixed constant $\rho > 1$?

Acknowledgments. Shubhang Kulkarni would like to thank Kishen Gowda for engaging in preliminary discussions about approximations for feedback vertex set and pseudoforest deletion set. These turned out to be a starting point for us in realizing the connections to submodular cover implicitly known in literature. We thank an anonymous reviewer for pointing out an error in our proof of a previous version of Theorem 1.3.

References

[1] V. Bafna, P. Berman, and T. Fujito. Constant ratio approximations of the weighted feedback vertex set problem for undirected graphs. In *Algorithms and Computations*, pages 142–151, 1995.

[2] C. Bazgan, A. Nichterlein, and S. Vazquez Alferez. Destroying Densest Subgraphs Is Hard. In *19th Scandinavian Symposium and Workshops on Algorithm Theory (SWAT 2024)*, pages 6:1–6:17, 2024.

[3] A. Becker and D. 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.

[4] D. Boob, Y. Gao, R. Peng, S. Sawlani, C. Tsourakakis, D. Wang, and J. Wang. Flowless: Extracting densest subgraphs without flow computations. In *Proceedings of The Web Conference 2020*, pages 573–583, 2020.

[5] K. Chandrasekaran, C. Chekuri, S. Fiorini, S. Kulkarni, and S. Weltge. Polyhedral aspects of feedback vertex set and pseudoforest deletion set. *Mathematical Programming*, pages 1–48, 2025.

[6] M. Charikar. Greedy Approximation Algorithms for Finding Dense Components in a Graph. In *Approximation Algorithms for Combinatorial Optimization*, pages 84–95, 2000.

[7] C. Chekuri, A. B. Christiansen, J. Holm, I. van der Hoog, K. Quanrud, E. Rotenberg, and C. Schwiegelshohn. Adaptive out-orientations with applications. In *Proceedings of the 2024 Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 3062–3088, 2024.

[8] C. Chekuri, K. Quanrud, and M. 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.

[9] F. A. Chudak, M. X. Goemans, D. S. Hochbaum, and D. 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.

[10] W. H. Cunningham. Optimal attack and reinforcement of a network. *Journal of the ACM (JACM)*, 32(3):549–561, 1985.

[11] L. Dhulipala, Q. C. Liu, S. Raskhodnikova, J. Shi, J. Shun, and S. 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.

[12] M. Dinitz, S. Kale, S. Lattanzi, and S. 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.

[13] I. Dinur and D. Steurer. Analytical approach to parallel repetition. In *Proceedings of the Forty-Sixth Annual ACM Symposium on Theory of Computing*, STOC, page 624–633, 2014.

[14] U. Feige. A threshold of ln n for approximating set cover. *Journal of the ACM (JACM)*, 45(4):634–652, 1998.

[15] A. Frank. *Connections in combinatorial optimization.* Oxford University Press, Oxford, 2011.

[16] S. Fujishige. Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research*, 5(2):186–196, 1980.

[17] T. Fujito. Approximating node-deletion problems for matroidal properties. *Journal of Algorithms*, 31(1):211–227, 1999.

[18] A. V. Goldberg. Finding a maximum density subgraph. Technical report, University of California at Berkeley, USA, 1984.

[19] E. Harb, K. Quanrud, and C. Chekuri. Faster and scalable algorithms for densest subgraph and decomposition. In *Advances in Neural Information Processing Systems*, 2022.

[20] E. Harb, K. Quanrud, and C. 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.

[21] S. Khot and O. Regev. Vertex cover might be hard to approximate to within 2- ε. *Journal of Computer and System Sciences*, 74(3):335–349, 2008.

[22] T. Lanciano, A. Miyauchi, A. Fazzone, and F. Bonchi. A survey on the densest subgraph problem and its variants. *ACM Comput. Surv.*, 56(8), Apr. 2024.

[23] A. Lee and I. Streinu. Pebble game algorithms and sparse graphs. *Discrete Mathematics*, 308(8):1425–1437, Apr. 2008.

[24] J. M. Lewis and M. Yannakakis. The node-deletion problem for hereditary properties is np-complete. *Journal of Computer and System Sciences*, 20(2):219–230, Apr. 1980.

[25] T. D. Nguyen and A. 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.

[26] J.-C. Picard and M. Queyranne. A network flow solution to some nonlinear 0-1 programming problems, with applications to graph theory. *Networks*, 12(2):141–159, 1982.

[27] S. Sawlani and J. Wang. Near-optimal fully dynamic densest subgraph. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 181–193, 2020.

[28] S. Ueno, Y. Kajitani, and S. 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, Dec. 1988.

[29] N. Veldt, A. R. Benson, and J. Kleinberg. The generalized mean densest subgraph problem. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1604–1614, 2021.

[30] D. J. Welsh. *Matroid Theory*, volume 8 of *LMS Monographs*. Academic Press, London, 1976.

[31] L. A. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

[32] M. 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.

A Reductions Between SupmodDD, MatroidFVS, and GraphDD

In this section, we show reductions between SupmodDD, MatroidFVS and GraphDD. Throughout the section, we use $b_G(Z) := \cup_{u \in Z} \delta_G(u)$ to denote the *edge-coverage* of a set of vertices $Z \subseteq V$ in a graph $G = (V, E)$. Furthermore, for a matroid $\mathcal{M} = (E, \mathcal{I})$, we use $\texttt{rank}_{\mathcal{M}}$ to denote its rank function, and $\mathcal{M}^*$ to denote the dual matroid. We first show that MatroidFVS is a special case of SupmodDD.

Theorem A.1. *Let $G = (V, E)$ be a graph and $\mathcal{M} = (E, \mathcal{I})$ be a matroid. Then, there exists a normalized, non-negative, integer-valued, supermodular function $f : 2^V \to \mathbb{Z}_{\geq 0}$ such that for a subset $F \subseteq V$ of vertices, we have that $E[V - F] \in \mathcal{I}$ if and only if $\lambda^*_{f|_{V-F}} \leq 1$.*

Proof. TOPROVE 10 □

Next, we show that for all integer $\rho \in \mathbb{Z}_+$, ρ-GraphDD is a special case of MatroidFVS. For this, we will need the following background. We recall that for an integer $\rho \in \mathbb{Z}_+$, the ρ-fold union of a matroid $\mathcal{M} = (E, \mathcal{I}_\rho)$ is another matroid $\mathcal{M}_\rho = (E, \mathcal{I}_\rho)$, where a subset of edges $F \subseteq E$ is in $\mathcal{I}_\rho$ if F can be partitioned into ρ parts such that each part is in $\mathcal{I}$, i.e., $F := \uplus_{i \in [\rho]} F^{(i)}$ such that $F^{(i)} \in \mathcal{I}$ for every $i \in [\rho]$. We refer the reader to Welsh's book on matroid theory [30] for additional details. We will also rely on a well-known characterization of *pseudoforests* using density—we recall that a graph is a pseudoforest if every component has at most one cycle. The following proposition states that pseudoforests are the graphs that have density at most 1.

Proposition A.1. *[5] A graph G is a pseudoforest if and only if $\lambda^*_G \leq 1$.*

We now show the connection between ρ-GraphDD and MatroidFVS when $\rho \in \mathbb{Z}_+$.

Theorem A.2. *Let $G = (V, E)$ be a graph and $\rho \in \mathbb{Z}_+$ be an integer. Let $\mathcal{M} := (E, \mathcal{I})$ denote the pseudoforest matroid on the graph $G = (V, E)$, where a set of edges $E' \subseteq E$ is independent if every component in the subgraph $G' = (V, E')$ has at most one cycle. Let $\mathcal{M}_\rho := (E, \mathcal{I}_\rho)$ be the ρ-fold union of the pseudoforest matroid. Then, for a subset $F \subseteq V$, we have that $E[V - F] \in \mathcal{I}_\rho$ if and only if $\lambda^*_{G-F} \leq \rho$.*

Proof. TOPROVE 11 □

Remark A.1. *There are several ways to prove Theorem A.2. For example, one can directly show that $\mathcal{M} = (E, \mathcal{I} := \{E' \subseteq E : \lambda^*_{(V,E')} \leq \rho\})$ is a matroid by showing that it satisfies the matroid axioms or by applying known results in submodularity and matroid theory (see Corollary 8.1 of [30]).*