

SDP-based Algorithms for Maximum Independent Set Problems on Hypergraphs

Geir Agnarsson¹, Magnús M. Halldórsson², and Elena Losievskaja³

¹ Dept. of Mathematical Sciences, George Mason University, Fairfax, VA, USA,
geir@math.gmu.edu,

² School of Computer Science, Reykjavík University, IS-103 Reykjavik, Iceland
mmh@ru.is,

³ Faculty of Industrial-, Mechanical Engineering and Computer Science, University of
Iceland, IS-107 Reykjavik, Iceland elenal@hi.is,

Abstract. This paper deals with approximations of maximum independent sets in non-uniform hypergraphs of low degree. We obtain the first performance ratio that is sublinear in terms of the maximum or average degree of the hypergraph. We extend this to the weighted case and give a $O(\bar{D} \log \log \bar{D} / \log \bar{D})$ bound, where \bar{D} is the average weighted degree in a hypergraph, matching the best bounds known for the special case of graphs. Our approach is to use a semi-definite technique to sparsify a given hypergraph and then apply combinatorial algorithms to find a large independent set in the resulting sparser instance.

1 Introduction

This paper deals with approximations of maximum independent sets in hypergraphs of low degree. Recall that a hypergraph (set system) $H = (V, E)$ has a vertex set V and a collection E of (*hyper*)edges that are arbitrary subsets of V . A hypergraph is weighted if vertices in V are assigned weights. It has *rank* r if all edges are of size at most r , and is *r-uniform* if all are of size exactly r . A set of vertices is *independent* if it does not properly contain any edge in E . The *degree* of a vertex is its number of incident edges. We consider approximation algorithms for the maximum independent set (MIS) problem in sparse non-uniform hypergraphs.

The MIS problem is of fundamental interest, capturing conflict-free sets in a very general way. It generalizes the classic independent set problem in graphs, and thus inherits all its hardness properties. The vertices not in an independent set form a hitting set of the hypergraph. Algorithms for MIS can therefore be viewed as set covering algorithms with a differential measure, which lends it an additional interest.

Hypergraph problems tend to be more difficult to resolve than the corresponding graph problems, with the MIS problem a typical case. The best performance ratio known for MIS in general hypergraphs, in terms of the number n of vertices, is only $O(n/\log n)$, which has a rather trivial argument [7]. For the graph case, for comparison, the ratio is $O(n(\log \log n)^2 / \log^3 n)$ [4]. In terms

of the maximum degree Δ , a ratio of Δ is trivial, while previous work on MIS in hypergraphs has improved only the constant term [2, 8]. More specifically, a $\Delta/1.365$ -upper bound was obtained for a greedy algorithm and a tight $(\Delta+1)/2$ -ratio for a local search method, while in [8] a tight bound of $(\Delta+1)/2$ was obtained for the greedy algorithm as well as the best previously known bound of $(\Delta+3)/5$. The main sign of success has been on sparse hypergraphs, where Turan-like bounds have been proven [3, 16, 15]. Unlike graphs, however, the exact constant in the bounds is not known.

The most powerful approach for the approximation of challenging optimization problems has involved the use of semi-definite programming (SDP). It is responsible for the best ratio known for IS in graphs of $O(\Delta \log \log \Delta / \log \Delta)$ [6]. It is also involved in the complementary vertex cover problem [9], both in graphs and in hypergraphs. Yet, it has failed to yield much success for MIS in hypergraphs, except for some special cases. One intuition may be that hyperedges result in significantly weaker constraints in the semi-definite relaxation than the graph edges. The special cases where it has been successful — 2-colorable k -uniform hypergraphs [6] and 3-uniform hypergraphs with a huge independence number [11] — have properties that result in strengthened constraints. The usefulness of SDP for general MIS has remained open.

This state-of-the-art suggests several directions and research questions. A key question is to what extent approximation ratios for IS in graphs can be matched in hypergraphs. This can be asked in terms of different degree parameters, as well as extensions. Given that graphs are 2-uniform hypergraphs and k -uniform hypergraphs have certain nice properties, the question is also how well we can handle non-uniform hypergraphs.

Our results. We derive the first $o(\Delta)$ -approximation for IS in hypergraphs matching the $O(\Delta \log \log \Delta / \log \Delta)$ -approximation for the special case of graphs. Our approach is to use an SDP formulation to sparsify the part of the instance formed by 2-edges (edges of size 2), followed by a combinatorial algorithm on the resulting sparser instance. This is extended to obtain an identical bound in terms of the average degree \bar{d} of an unweighted hypergraph. As part of the method, we also obtain a $k^{5/2-1/k} \bar{d}^{1-1/k+o(1)}$ -approximation for hypergraphs with independence number at least n/k .

We generalize the results to the vertex-weighted problem. In that case, no non-trivial bound is possible in terms of the average degree alone, so we turn our attention to a weighted version. The *average weighted degree* \bar{D} is the node-weighted average of the vertex degrees. We give a $O(\bar{D} \log \log \bar{D} / \log \bar{D})$ -approximation for MIS.

We apply two combinatorial algorithms to hypergraphs with few 2-edges. One is a greedy algorithm analyzed by Caro and Tuza [3] for the k -uniform case and Thiele [16] for the non-uniform case. The bound obtained in [16] is in general unwieldy, but we can show that it gives a good approximation when the number of 2-edges has been reduced. The other is a simple randomized algorithm analyzed by Shachnai and Srinivasan [15].

Organization. The paper is organized in the following way. In Sect. 2 we describe how to find a large sparse hypergraph in a given hypergraph H using SDP technique. In Sect. 3 we give analysis of the greedy algorithm for MIS on hypergraphs of rank 3 with small 2-degree, and then show how to apply this greedy algorithm together with SDP to find a large hypergraph in H . In Sect. 4 we describe how to use a randomized algorithm together with SDP to find a good approximation of a weighted MIS in hypergraphs.

2 Definitions

Given a hypergraph $H = (V, E)$, let n and m be the number of vertices and edges in H , respectively. We assume that H is a *simple* hypergraph, i.e. no edge is a proper subset of another edge. An edge of size t is a t -edge. A hypergraph is r -uniform if all edges are r -edges. A *graph* is a 2-uniform hypergraph. The *rank* r of a hypergraph H is the maximum edge size in H .

Let $d_t(v)$ be t -degree of a vertex v , or the number of t -edges incident on v . We denote by Δ_t and \bar{d}_t the maximum and the average t -degree in a hypergraph, respectively. The *degree* $d(v)$ of a vertex v is the total number of edges incident on v , i.e. $d(v) = \sum_{t=2}^r d_t(v)$. We denote by Δ and \bar{d} the maximum and the average degree in a hypergraph, respectively.

Given a function $f : V \rightarrow R$ that assigns weights to the vertices of H , let $w(H) = w(V) = \sum_{v \in V} w(v)$. We define $D(v) = w(v)d(v)$ and $\bar{D} = \sum_{v \in V} w(v)d(v) / \sum_{v \in V} w(v)$ to be the *weighted degree* of a vertex v and the *average weighted degree* in H , respectively.

By *deleting a vertex v from a hypergraph H* we mean the operation of deleting v and all incident edges from H . By *deleting a vertex v from an edge e* we mean the operation of replacing e by $e \setminus \{v\}$.

A (weak) *independent set* in H is a subset of V that doesn't properly contain any edge of H . Let $\alpha(H, w)$ be the weight of a maximum independent set in H . If H is unweighted, then it is denoted as $\alpha(H)$.

3 Semidefinite Programming

We use semidefinite programming to find large subgraphs with few 2-edges. More generally, we find subgraphs of large weight and small weighted average degree. This is obtained by rounding the vector representation of a suitable subgraph; such a subgraph is found by a result of Alon-Kahale. Along the way, we twice eliminate vertices of high-degree to ensure degree properties.

Let us recall the definition of a vector coloring of a graph [10].

Definition 3.1 ([10]). *Given a graph G and a real number $h \geq 1$, a vector h -coloring of G is an assignment of a $|V(G)|$ -dimensional unit vector v_i to each*

vertex v_i of G so that for any pair v_i, v_j of adjacent vertices the inner product of their vectors satisfies

$$\mathbf{v}_i \cdot \mathbf{v}_j \leq -\frac{1}{h-1}. \quad (1)$$

The *vector chromatic number* $\bar{\chi}(G)$ is the smallest positive number h , such that there exists a feasible vector h -coloring of G .

A vector representation given by a vector coloring is used to find a sparse subgraph by the means of *vector rounding* [10]: choose a random vector \mathbf{b} , and retain all vertex vectors whose inner product with \mathbf{b} is above a certain threshold. The quality (i.e. sparsity) of the rounded subgraph depends on the vector chromatic number of the graph. In order to approximate independent sets we need to use this on graphs that do not necessarily have a small vector chromatic number but have a large independent set.

A graph with a large independent set contains a large subgraph with a small vector chromatic number, and there is a polynomial time algorithm to find it. This comes from the following variation of a result due to Alon and Kahale [1]:

Theorem 3.2 ([7]). *Let $G = (V, E, w)$ be a weighted graph and ℓ, p be numbers such that $\alpha(G, w) \geq w(G)/\ell + p$. Then, there is a polynomial time algorithm that gives an induced subgraph G_1 in G with $w(G_1) \geq p$ and $\bar{\chi}(G_1) \leq \ell$.*

Let us now present our algorithm for finding a large-weight low 2-degree hypergraph. It assumes that it is given the size α of the maximum independent set in the graph. We can sidestep that by trying all possible values for α , up to a sufficient precision (say, factor 2).

Algorithm SPARSEHYPERGRAPH

Input: Hypergraph $H(V, E)$, and its independence number α

Output: Induced hypergraph \hat{H} in H of maximum degree $2k\bar{D}$ and maximum 2-degree $\sqrt{2k\bar{D}}$

Let $k = w(H)/\alpha$ and $a = 1 + \frac{1}{\ln \ln D - 1}$.

Let G be the graph induced by the 2-edges of H .

Let G_0 be the subgraph of G induced by nodes of degree at most $2k\bar{D}$ in H .

Find an induced subgraph G_1 in G_0 with $w(G_1) \geq \frac{(a-1)w(G)}{2ak}$
with a vector $2ak$ -coloring.

Choose a random $|V(G_1)|$ -dimensional vector \mathbf{b} .

Let G_2 be the subgraph of G_1 induced by vertices $\{v \in V(G_1) : \mathbf{v} \cdot \mathbf{b} \geq c\}$,

where $c = \sqrt{\frac{ak-2}{ak} \ln(2k\bar{D})}$.

Let \hat{V} be the set of vertices of degree at most $\sqrt{2k\bar{D}}$ in G_2 .

Output \hat{H} , the subhypergraph in H induced by \hat{V} .

Fig. 1. The sparsifying algorithm

The algorithm SPARSEHYPERGRAPH can be implemented to run in polynomial time. The subgraph G_1 in G_0 with small vector chromatic number and large independent set can be found in polynomial time [1]. A vector representation can be found within an additive error of ϵ in time polynomial in $\ln 1/\epsilon$ and n using the ellipsoid method [5] and Choleski decomposition.

Analysis

Lemma 3.3. *The graph G_0 has weight at least $w(H)(1 - 1/2k)$ and independence number at least $\alpha(H)/2$.*

Proof. The graph G has the same weight as H , or $w(V)$. The independence number of G is also at least that of H , since it contains only a subset of the edges. Let $X = V(G) - V(G_0)$ be the high-degree vertices deleted to obtain G_0 . Then,

$$\sum_{v \in X} w(v)d(v) \geq \sum_{v \in X} w(v) \cdot 2k\bar{D} = 2k\bar{D}w(X) . \quad (2)$$

Since

$$\bar{D} \cdot w(V) = \sum_{v \in V} w(v)d(v) \geq \sum_{v \in X} w(v)d(v) , \quad (3)$$

we get from combining (3) with (2) that the weight $w(X)$ of the deleted vertices is at most $w(V)/2k$. Thus, $w(G) \geq (1 - 1/2k)w(H)$. Also, G_0 has a maximum independent set of weight at least $\alpha(G_0, w) \geq \alpha(G, w) - w(X) \geq \alpha(H, w) - w(X) \geq w(H)/2k$.

Observe that $\alpha(G_0, w) \geq w(H)/2k \geq w(G_0)/2k = w(G_0)/2ak + w(G_0)(a - 1)/2ak$. Then, Theorem 3.2 ensures that a subgraph G_1 can be found with $w(G_1) \geq w(G_0)(a - 1)/2ak$ and $\bar{\chi}(G_1) \leq 2ak$. From that, a vector $2ak$ -coloring of G_1 can be found.

The main task is to bound the properties of the rounded subgraph G_2 . Karger et al. [10] estimated the probability that G_2 contains a given vertex or an edge. Let $N(x)$ denote the tail of the standard normal distribution: $N(x) = \int_x^\infty \phi(z)dz$, where $\phi(z) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{z^2}{2}\right)$ is the density function. Let $\tau = \sqrt{\frac{2(ak-1)}{ak-2}}$.

Lemma 3.4 ([10]). *A graph G_2 induced in $G_1(V_1, E_1)$ after vector-rounding contains a given vertex in V_1 with probability $N(c)$ and a given edge in E_1 with probability $N(c\tau)$.*

The following lemma states well-known bounds on the tail of the normal distribution.

Lemma 3.5 ([14]). *For every $x > 0$, $\phi(x)\left(\frac{1}{x} - \frac{1}{x^3}\right) < N(x) < \phi(x)\frac{1}{x}$.*

We can now bound the weight of the subgraph found.

Lemma 3.6. *\hat{V} has expected weight $\Omega\left(\frac{w(G_1)}{(k\bar{D})^{\frac{1}{2} - \frac{1}{k}} \sqrt{\ln(k\bar{D})}}\right)$. This can be derandomized to obtain an induced subgraph \hat{V} with this much weight and maximum 2-degree at most $\sqrt{2k\bar{D}}$.*

Proof. First, for any edge (u, v) in G_1 and G_2 we define a weight function $w(u, v) = w(u) + w(v)$. Let $w(V_1) = \sum_{v \in V(G_1)} w(v)$ and $w(E_1) = \sum_{(v, u) \in E(G_1)} (w(v) + w(u))$ be the weight of vertices and edges in G_1 . Similarly, let $w(V_2)$ and $w(E_2)$ be the weight of vertices and edges in G_2 . Let X_i be an indicator random variable with $X_i = 1$, if V_2 contains $v_i \in V_1$ and $X_i = 0$ otherwise. Then, $w(V_2) = \sum_{v_i \in V_1} w(v_i) X_i$. Using Lemma 3.4 and linearity of expectation we bound $E[w(V_2)]$ by

$$E[w(V_2)] = w(V_1)N(c) . \quad (4)$$

Similarly, we bound $E[w(E_2)]$ by

$$E[w(E_2)] = w(E_1)N(c\tau) \leq 2k\bar{D}w(V_1)N(c\tau) , \quad (5)$$

where in the last inequality we use the fact that maximum degree in G_1 is bounded by $2k\bar{D}$ (since we deleted the high-degree vertices from G and G_1 is an induced subgraph in G). Combining (4) and (5), we get that

$$E \left[w(V_2) - \frac{w(E_2)}{\sqrt{2k\bar{D}}} \right] = w(V_1)N(c) - \sqrt{2k\bar{D}}w(V_1)N(c\tau) . \quad (6)$$

Observe that

$$c\tau = \sqrt{\frac{2(ak-1)}{ak} \ln(2k\bar{D})} = \sqrt{2 \left(1 - \frac{1}{ak}\right) \ln(2k\bar{D})}$$

and

$$\exp(-(c\tau)^2/2) = (2k\bar{D})^{-1+1/ak} .$$

Then, by Lemma 3.5

$$N(c\tau) < \phi(c\tau) \frac{1}{c\tau} = \frac{(2k\bar{D})^{-1+1/ak}}{\sqrt{2\pi} \cdot \sqrt{\frac{2(ak-1)}{ak} \ln(2k\bar{D})}} \quad (7)$$

and

$$N(c) > \phi(c) \frac{1}{c} \left(1 - \frac{1}{c^2}\right) = \frac{(2k\bar{D})^{-1/2+1/ak}}{\sqrt{2\pi} \cdot \sqrt{\frac{ak-2}{ak} \ln(2k\bar{D})}} \left(1 - \frac{ak}{(ak-2) \ln(2k\bar{D})}\right) . \quad (8)$$

Combining (6), (7) and (8), we deduce that

$$\begin{aligned} E \left[w(V_2) - \frac{w(E_2)}{\sqrt{2k\bar{D}}} \right] &> w(V_1) \frac{(2k\bar{D})^{-1/2+1/ak}}{\sqrt{2\pi} \cdot \sqrt{\frac{ak-2}{ak} \ln(2k\bar{D})}} \left(1 - \frac{ak}{(ak-2) \ln(2k\bar{D})} - \sqrt{\frac{ak-2}{2(ak-1)}}\right) \\ &= \Omega \left(\frac{w(V_1)}{(k\bar{D})^{1/2-1/k} \sqrt{\ln(k\bar{D})}} \right) , \end{aligned} \quad (9)$$

where in the last line we use $a = 1 + \frac{1}{\ln \ln D - 1}$.

The weight of vertices with degree greater than $\sqrt{2k\bar{D}}$ is at most $\sum_{v_i \in V_2} w(v_i)d(v_i)/\sqrt{2k\bar{D}} = w(E_2)/\sqrt{2k\bar{D}}$. After deleting all such vertices from G_2 , the expected weight of \hat{V} is $E \left[w(V_2) - \frac{w(E_2)}{\sqrt{2k\bar{D}}} \right]$ which is bounded by (9).

Finally, we can apply derandomization technique from [13] to derandomize the vector rounding in polynomial time. In our algorithm an elementary event corresponds to an edge in G_2 and involves only two vectors corresponding to the endpoints of the edge. This completes the proof.

We can bound the weight of the resulting hypergraph \hat{H} in terms of the original hypergraph. We have that $w(\hat{V}) = \Omega \left(\frac{w(G_1)}{(k\bar{D})^{\frac{1}{2} - \frac{1}{k}} \sqrt{\ln k\bar{D}}} \right)$, while using $a = 1 + \frac{1}{\ln \ln D - 1}$, we have that

$$w(G_1) = \frac{(a-1)w(G)}{2ak} = \frac{w(G)}{2k \ln \ln \bar{D}} = \frac{w(H) \left(1 - \frac{1}{2k}\right)}{2k \ln \ln \bar{D}} = \Omega \left(\frac{w(H)}{k \ln \ln \bar{D}} \right).$$

Theorem 3.7. *Let H be a hypergraph with average weighted degree \bar{D} . The SPARSEHYPERGRAPH algorithm finds an induced hypergraph in H of weight $\Omega \left(\frac{w(H)}{k^{3/2-1/k} \bar{D}^{1/2-1/k} \ln \ln \bar{D} \sqrt{\ln(k\bar{D})}} \right)$, maximum 2-degree at most $\sqrt{2k\bar{D}}$, and maximum degree at most $2k\bar{D}$.*

4 Greedy Algorithm

Given a hypergraph H on n vertices with average degree \bar{d} , our GREEDYSDP algorithm first finds a sparse induced hypergraph H' in H using the SPARSEHYPERGRAPH algorithm and then uses the GREEDY algorithm to find an independent set in H' .

The GREEDY algorithm is a natural extension of the max-degree greedy algorithm on graphs and uniform hypergraphs and was analyzed by Thiele [16]. Given a hypergraph $H(V, E)$ with rank r , for any vertex $v \in V$ let $\vec{d}(v) = (d_1(v), \dots, d_r(v))$ be the *degree vector* of v , where $d_i(v)$ is the number of edges of size i incident on v . Then, for any vertex $v \in V$ let

$$f(\vec{d}(v)) = \sum_{i_1}^{d_1(v)} \sum_{i_2}^{d_2(v)} \cdots \sum_{i_r}^{d_r(v)} \prod \binom{d_1}{i_1} \prod \binom{d_2}{i_2} \cdots \prod \binom{d_r}{i_r} \frac{(-1)^{\sum_{j=1}^r i_j}}{\sum_{j=1}^r (j-1)i_j + 1}$$

and let $F(H) = \sum_{v \in V} f(\vec{d}(v))$. The GREEDY algorithm iteratively chooses a vertex $v \in V$ with $F(H \setminus v) \geq F(H)$ and deletes v with all incident edges from H until the edge set is empty. The remaining vertices form an independent set in H .

Caro and Tuza [3] showed that in r -uniform hypergraphs the GREEDY algorithm always finds a weak independent set of size at least $\Theta\left(n/\Delta^{\frac{1}{r-1}}\right)$. Thiele [16] extended their result to non-uniform hypergraphs and gave a lower bound on the independence number as a complicated function of the degree vectors of the vertices in a hypergraph. Using these two bounds, we prove the following lemma.

Lemma 4.1. *Given a hypergraph H on n vertices with maximum 2-degree \sqrt{d} and maximum degree d , the GREEDY algorithm finds an independent set of size $\Omega(n/\sqrt{d})$.*

Proof. First, we truncate edges in H to a maximum size three by arbitrarily deleting excess vertices. The resulting hypergraph H' still has maximum 3-degree d and maximum 2-degree \sqrt{d} , and is now of rank 3. Moreover, an independent set in H' is also independent in H . Thus, to prove the claim it is sufficient to bound from below the size of an independent set found by the greedy algorithm in H' .

As shown in [16], GREEDY finds an independent set in a rank-3 hypergraph of size at least

$$\alpha(H') \geq n \sum_{j=0}^d \sum_{i=0}^{\sqrt{d}} \binom{d}{j} \binom{\sqrt{d}}{i} \frac{(-1)^{j+i}}{i+2j+1}. \quad (10)$$

By using the equality $\sum_k \binom{n}{k} \frac{(-1)^k}{x+k} = x^{-1} \binom{x+n}{n}^{-1}$ we can simplify (10) as:

$$\begin{aligned} \alpha(H') &\geq n \sum_{i=0}^{\sqrt{d}} (-1)^i \binom{\sqrt{d}}{i} \frac{1}{2} \left(\sum_{j=0}^d \binom{d}{j} \frac{(-1)^j}{j+(i+1)/2} \right) \\ &= \frac{n}{2(d+1)} \sum_{i=0}^{\sqrt{d}} (-1)^i \binom{\sqrt{d}}{i} \binom{(i+1)/2+d}{d+1}^{-1}. \end{aligned} \quad (11)$$

We show that for any value of d

$$F_d = \sum_{i=0}^{\sqrt{d}} (-1)^i \binom{\sqrt{d}}{i} \binom{(i+1)/2+d}{d+1}^{-1} \quad (12)$$

is lower bounded by $x\sqrt{d}$ for some $x > 0$. Then, from (11) the GREEDY algorithm finds an independent set of size at least $\Omega(n/\sqrt{d})$.

Let $f_d(i) = \binom{\sqrt{d}}{i} \binom{(i+1)/2+d}{d+1}^{-1}$. Abusing binomial notation, we assume that $\binom{\sqrt{d}}{i} = 0$, for any $i > \sqrt{d}$ and \sqrt{d} integral. Then,

$$F_d = \sum_{i=0}^{\sqrt{d}} (-1)^i f_d(i). \quad (13)$$

We define

$$q_d(i) = \frac{(i+2)(i+4)\cdots(i+2d+2)}{(i+3)(i+5)\cdots(i+2d+1)} \quad (14)$$

for any $i \geq 0$. Using Stirling's approximation for the factorial function⁴ we obtain

$$q_d(0) = \frac{2^{2d+1}(d+1)d}{(2d+1)} = \sqrt{\pi d} \left(1 + O\left(\frac{1}{d}\right)\right)$$

and

$$q_d(1) = \frac{(2d+3)}{2^{2d+1}(d+1)(d+1)} = \sqrt{\frac{d}{\pi}} \left(1 + O\left(\frac{1}{d}\right)\right) .$$

Note that $q_d(i+2) = \frac{(i+3)(i+2d+4)}{(i+2)(i+2d+3)}q_d(i) > q_d(i)$, and so $q_d(i) > \sqrt{d}$ for any $i \geq 0$.

Then, from the definition of $f_d(i)$ and (14) we have that $\frac{f_d(i+1)}{f_d(i)} = \frac{\sqrt{d-i}}{q_d(i)} < 1$. From (12) and (13) it follows that $F_d > f_d(0) - f_d(1)$ and $f_d(0) = q_d(0)$, then

$$\begin{aligned} F_d &> f_d(0) - f_d(1) \\ &= f_d(0) \left(1 - \frac{\sqrt{d}}{q_d(0)}\right) \\ &= q_d(0) - \sqrt{d} \\ &= (\sqrt{\pi} - 1)\sqrt{d} \left(1 + O\left(\frac{1}{d}\right)\right) . \end{aligned} \quad (15)$$

Thus, from (11), (12) and (15) the GREEDY algorithm finds an independent set of size at least $\Omega(n/\sqrt{d})$.

The bound on the performance ratio of GREEDYSDP then follows from Lemma 4.1, Theorem 3.7 and the fact that truncating edges in SPARSEHYPERGRAPH doesn't increase the weight of a maximal independent set in a hypergraph.

Theorem 4.2. *Given a hypergraph H on n vertices with average degree \bar{d} and the independence number $\alpha(H) = n/k$, the GREEDYSDP algorithm finds an independent set of size at least $\Omega\left(\frac{n}{k^{5/2-1/k}\bar{d}^{1-1/k}\ln\ln\bar{d}\sqrt{\ln(k\bar{d})}}\right)$.*

From Theorem 4.2 it is easy to see that if the maximum independent set in H is relatively big, say $\Omega\left(\frac{n\ln\ln\bar{d}}{\ln d}\right)$, i.e. $k = O\left(\frac{\ln\bar{d}}{\ln\ln d}\right)$, then GREEDYSDP obtains an approximation ratio of $O\left(\frac{\bar{d}}{\ln d}\right)$. However, if the maximum independent set is at most $\Omega\left(\frac{n\ln\ln\bar{d}}{\ln d}\right)$, then GREEDY alone is within a factor of $O\left(\frac{\bar{d}\ln\ln\bar{d}}{\ln d}\right)$. Therefore, we run both GREEDY and GREEDYSDP and output the larger independent set found.

⁴ Stirling's approximation: $N! = \sqrt{2\pi N} \left(\frac{N}{e}\right)^N \left(1 + O\left(\frac{1}{N}\right)\right)$

Theorem 4.3. *Given a hypergraph H with average degree \bar{d} , the GREEDYSDP-MIS approximates the maximum independent set within a factor of $O\left(\frac{\bar{d} \ln \ln \bar{d}}{\ln \bar{d}}\right)$.*

Corollary 4.4. *For small k the approximation factor of GREEDYSDP-MIS is $O\left(\bar{d}^{1-\frac{1}{k}+o(1)}\right)$.*

The implication is the first approximation factor for the independent set problem in hypergraphs that is sublinear in the average degree.

Corollary 4.5. *The independent set problem in hypergraphs is $o(\Delta)$ -approximable.*

5 Randomized Algorithm

The RANDOMIS algorithm extends the randomized version of Turán bound on graphs and was analyzed by Shachnai and Srinivasan in [15]. Given a hypergraph $H(V, E)$, the algorithm creates a random permutation π of V and adds a vertex v to the independent set I , if there is no edge e such that e contains v and v appears last in π among the vertices of e . Clearly, RANDOMIS outputs a feasible independent set I , since it never contains the last vertex in any edge under the permutation π .

Shachnai and Srinivasan [15] analyzed RANDOMIS on weighted hypergraphs. They gave a lower bound on the probability that a vertex $v \in H$ is added by the algorithm to the independent set, using conditional probabilities and the FKG inequality. In uniform hypergraphs the lower bound on the size of a independent set found by RANDOMIS follows by summing the probabilities over the vertices and applying linearity of expectation, giving a bound identical to that of Caro and Tuza [3].

Theorem 5.1 ([15], Theorem 2). *For any $k \geq 2$ and any k -uniform hypergraph H , RANDOMIS finds an independent set of size at least $\sum_{v \in V} \binom{d(v)+1}{d(v)}^{k-1} = \Omega\left(\sum_{v \in V} \frac{w(v)}{(d(v))^{k-1}}\right)$.*

To extend the bound to non-uniform weighted hypergraphs, Shachnai and Srinivasan introduced the following potential function on a vertex v :

$$f(v) = \min_{j=1,2,\dots,a(v)} (d_j(v))^{-\frac{1}{k_j(v)-1}},$$

where a vertex v lies in edges of $a(v)$ different sizes: $k_j(v)$, for $j = 1, 2, \dots, a(v)$, and $d_j(v)$ is the number of edges of size $k_j(v)$. Using similar analysis as in Theorem 5.1, they proved the following bound:

Theorem 5.2 ([15], Theorem 3). *Given a weighted hypergraph $H(V, E)$, the expected weight of the independent set produced by RANDOMIS is at least $\Omega\left(\sum_{v \in V} \frac{w(v)}{a(v)^{1/b(v)}} f(v)\right)$, where $b(v) = (\min_j (k_j(v) - 1))$.*

Shachnai and Srinivasan also show in [15] how to derandomize RANDOMIS for hypergraphs with bounded maximum degree, or logarithmic degree and sparse neighborhoods.

Our algorithm RANDOMSDP first uses SPARSEHYPERGRAPH to find an induced hypergraph H' in H with maximum 2-degree at most $\sqrt{2k\bar{D}}$ and maximum degree at most $2k\bar{D}$; and then uses RANDOMIS to find an independent set in H' .

The bound on the performance ratio of RANDOMSDP follows from Theorem 3.7 and Theorem 5.2, using that $\Omega\left(\sum_{v \in V} \frac{w(v)}{a(v)^{1/b(v)}} f(v)\right) = \Omega\left(\sum_{v \in V} \frac{w(v)}{\sum_{i=2}^r d_i^{\frac{1}{i-1}}}\right)$ by the definitions of $a(v)$, $b(v)$ and $f(v)$.

Theorem 5.3. *Given a weighted hypergraph H with average weighted degree \bar{D} , the RANDOMSDP algorithm finds an independent set of weight at least $\Omega\left(\frac{w(H)}{k^{2-1/k} \bar{D}^{1/2-1/k} \ln \bar{D} \sqrt{\ln(k\bar{D})}}\right)$.*

From Theorem 5.3 it follows that the RANDOMSDP algorithm approximates MIS within a factor of $O\left(\frac{\bar{D}}{\ln \bar{D}}\right)$ if $\alpha(H, w) = \Omega\left(\frac{w(V) \ln \ln \bar{D}}{\ln \bar{D}}\right)$, whereas RANDOMIS alone finds an approximation within a factor of $O\left(\frac{\bar{D} \ln \ln \bar{D}}{\ln \bar{D}}\right)$ if $\alpha(H, w) = O\left(\frac{w(V) \ln \ln \bar{D}}{\ln \bar{D}}\right)$. Therefore, given a hypergraph H , we run both RANDOMIS and RANDOMSDP on H and output the larger of the independent sets.

Theorem 5.4. *Given a hypergraph $H(V, E)$ with average weighted degree \bar{D} , the RANDOMSDP-MIS approximates the weight of a maximum independent set in H within a factor of $O\left(\frac{\bar{D} \ln \ln \bar{D}}{\ln \bar{D}}\right)$.*

6 Conclusions

In this paper we propose a new approach to the Maximum Independent Set problem in weighted non-uniform hypergraphs. Our approach is to use SDP techniques to sparsify a given hypergraph and then apply a combinatorial algorithm to find a large independent set. Using this approach we derive $o(\bar{d})$ -approximation for IS in unweighted hypergraphs, matching the best known ratio for IS in graphs, both in terms of maximum and average degree. We generalize the results to weighted hypergraphs, proving similar bounds in terms of the average weighted degree \bar{D} .

For further work one possible direction is to extend the result on the GREEDYSDP-MIS to weighted hypergraphs. Another (and perhaps more interesting) open question is to prove similar bounds in terms of the maximum and average weighted *hyperdegree*, where the *hyperdegree* $d^*(v)$ of a vertex v is defined as $d^*(v) = \sum_{t=2}^r d_t(v)^{\frac{1}{t-1}}$. The hyperdegree is a generalization of a vertex degree in a graph.

References

1. Alon, N., Kahale, N.: Approximating the independence number via the θ -function. *Math. Prog.* 80(3), 253–264 (1998)
2. Bazgan, C., Monnot, J., Paschos, V., Serrière, F.: On the differential approximation of MIN SET COVER. *Theor. Comp. Science* 332, 497–513 (2005)
3. Caro, Y., Tuza, Z.: Improved lower bounds on k -independence. *J. Graph Theory* 15, 99–107 (1991)
4. Feige, U.: Approximating maximum clique by removing subgraphs. *SIAM J. Disc. Math.* 18(2), 219–225 (2005)
5. Grötschel, M., Lovász, L., Schrijver, A.: The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* 1(2), 169–197 (1981).
6. Halldórsson, M. M.: Approximations of independent sets in graphs. In: Jansen, K., Rolim, J. (eds.) *Proc. APPROX LNCS*, vol. 1444, pp. 113. Springer, Berlin (1998)
7. Halldórsson, M. M.: Approximations of Weighted Independent Set and hereditary subset problems. *J. Graph Algorithms and Applications*, 4(1), 1–16 (2000)
8. Halldórsson, M. M., Losievskaja, E.: Independent sets in bounded-degree hypergraphs. *Disc. Appl. Math.* 157, 1773–1786 (2009)
9. Halperin, E.: Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs. *SIAM J. Computing* 31(5), 1608–1623 (2002)
10. Karger, D., Motwani, R., Sudan, M.: Approximate graph coloring by semidefinite programming. *J. ACM* 45(2), 246–265 (1998)
11. Krivelevich, M., Nathaniel R., Sudakov, B.: Approximating coloring and maximum independent sets in 3-uniform hypergraphs. *J. Algorithms* 41(1), 99–113 (2001)
12. Lovász, L.: On Shannon capacity of a graph. *IEEE Trans. Inform. Theory* 25(1), 1–7 (1979)
13. Mahajan, S., Ramesh, H.: Derandomizing semidefinite programming based approximation algorithms. *SIAM J. Computing* 28(5), 1641–1663 (1999)
14. Rényi, A.: *Probability theory*. Elsevier, New York, (1970)
15. Shachnai, H., Srinivasan, A.: Finding large independent sets of hypergraphs in parallel. *SIAM J. Disc. Math.* 18(3), 488–500 (2005)
16. Thiele, T.: A lower bound on the independence number of arbitrary hypergraphs. *J. Graph Theory* 32, 241–249 (1999)