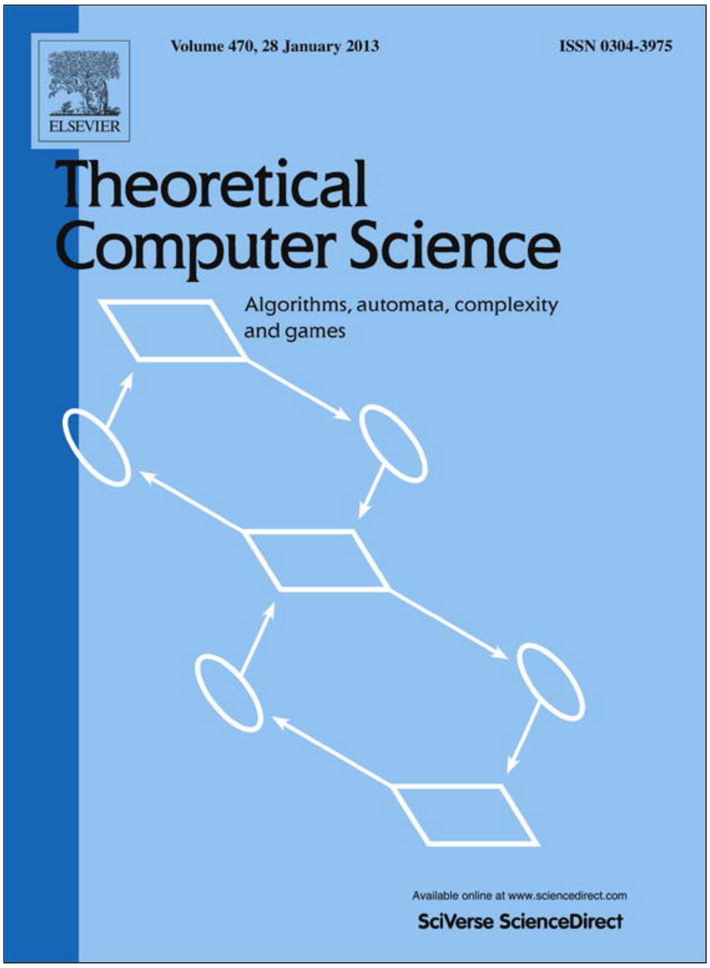
Volume 470, 28 January 2013
ISSN 0304-3975
ELSEVIER
Theoretical Computer Science
Algorithms, automata, complexity and games
Available online at www.sciencedirect.com
SciVerse ScienceDirect

Theoretical Computer Science 470 (2013) 1–9

Contents lists available at SciVerse ScienceDirect

Theoretical Computer Science

journal homepage: www.elsevier.com/locate/tcs

# SDP-based algorithms for maximum independent set problems on hypergraphs ☆

Geir Agnarsson [a], Magnús M. Halldórsson [b,*], Elena Losievskaja [b]

[a] *Department of Mathematical Sciences, George Mason University, Fairfax, VA, USA*
[b] *School of Computer Science, Reykjavik University, 101 Reykjavik, Iceland*

ARTICLE INFO

*Article history:*
Received 22 August 2011
Received in revised form 24 May 2012
Accepted 16 November 2012
Communicated by T. Erlebach

*Keywords:*
Independent sets
Hypergraphs
Approximation algorithms

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 an 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 [9]. For the graph case, for comparison, the ratio is $O(n(\log\log n)^2/\log^3 n)$ [6]. In terms of the maximum degree $\Delta$, a ratio of $\Delta$ is trivial, while previous work on MIS in hypergraphs has improved only the constant term [3,10]. 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 [10] 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 [5,18,17]. Unlike graphs, however, the exact constant in the bounds is not known.

☆ A preliminary version was presented at the 36th International Colloquium on Automata, Languages and Programming (ICALP), 2009. Research supported by Iceland Research Fund grant 70009022.

* Corresponding author. Tel.: +354 8256384.
*E-mail addresses:* geir@math.gmu.edu (G. Agnarsson), mmh@ru.is, magnusmh@gmail.com (M.M. Halldórsson), ellossie@gmail.com (E. Losievskaja).

0304-3975/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
doi:10.1016/j.tcs.2012.11.025

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 MIS in graphs of $O(\Delta \log\log \Delta / \log \Delta)$ [8]. It is also involved in the best approximations for the complementary problem of minimum vertex cover [11], 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 yield significantly weaker constraints in the semi-definite relaxation than graph edges. The special cases where it has been successful — 2-colorable $k$-uniform hypergraphs [8] and 3-uniform hypergraphs with a huge independence number [13] — have properties that result in strengthened constraints. The usefulness of SDP for general MIS has remained open.

It has recently been shown that it is NP-hardness of finding an independent set of size larger than $O(n(\frac{\log \Delta}{\Delta})^{\frac{1}{r-1}})$ in a 2-colorable $r$-uniform hypergraph for each fixed $r > 4$ [4]. In comparison, a simple greedy algorithm is known to find independent sets of size $\Omega(n(\frac{1}{\Delta})^{\frac{1}{r-1}})$ in any $r$-uniform hypergraph of maximum degree $\Delta$ [5]. It was also shown to be computationally hard to find an independent set of size more than $O(n\Delta^{-\frac{1}{r-1}} \log^{\frac{r}{r-1}} \Delta)$ in $r$-uniform hypergraphs that contain an independent set of size $n(1 - O(\log r/r))$, assuming the *Unique games* conjecture [4]. Finally, the MIS problem on graphs is hard to approximate within $\Omega(\Delta/\log^2 \Delta)$ assuming the unique games conjecture [2].

This state-of-the-art suggests several directions and research questions. A key question is to what extent approximation ratios for MIS 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 that $k$-uniform hypergraphs have certain nice properties, the question is also how well non-uniform hypergraphs can be handled.

*Our results.* We derive the first $o(\Delta)$-approximation for MIS 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 $k\bar{d}^{1-1/k+o(1)}$-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 [5] for the $r$-uniform case and Thiele [18] for the non-uniform case. The bound obtained in [18] 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 [17].

*Organization.* The paper is organized in the following way. In Section 2 we describe how to find a large sparse hypergraph in a given hypergraph $H$ using a semi-definite programming (SDP) technique. In Section 3 we analyze a 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 Section 4 we describe how to use a randomized algorithm together with SDP to find a good approximation of 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*. The *rank* $r$ of a hypergraph $H$ is the maximum edge size in $H$. A hypergraph is $r$-*uniform* if all edges are $r$-edges. A *graph* is a 2-uniform hypergraph.

Let $d_t(v)$ be $t$-*degree* of a vertex $v$, or the number of $t$-edges incident on $v$. We denote by $\Delta_t$ 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 $\Delta$ and $\bar{d}$ the maximum and the average degree in a hypergraph, respectively.

Given a function $w : V \to R$ 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

$$\bar{D} = \frac{\sum_{v \in V} w(v)d(v)}{w(V)}$$

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 $\alpha(H)$. We denote $k = w(H)/\alpha(H, w)$, for the hypergraph $H$ in question.

Algorithm SparseHypergraph
Input: Hypergraph $H(V, E)$, and its weighted independence number $\alpha$
Output: Induced hypergraph $\hat{H}$ in $H$ of maximum degree $2k\bar{D}$
and maximum 2-degree $\sqrt{2k\bar{D}}$, where $k = w(H)/\alpha(H, w) > 2$

Let $a = \frac{1}{1 - k/\log(k\bar{D})}$.
Let $G$ be the graph (on $V(H)$) formed by the 2-edges of $H$, and $G_0$ 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 $w(G_1) \geq \frac{(a-1)w(G_0)}{2ak}$ along with a vector $2ak$-coloring.
Choose a random $|V(G_1)|$-dimensional vector $\vec{b}$.
Let $G_2$ be the subgraph of $G_1$ induced by vertices $\{v \in V(G_1) : \vec{v} \cdot \vec{b} \geq c\}$, 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 $\hat{H} = H[\hat{V}]$, the subhypergraph in $H$ induced by $\hat{V}$.

**Fig. 1.** The sparsifying algorithm.

## 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. 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 [12] (See Fig. 1).

**Definition 3.1** (*[12]*)**.** 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 $\vec{v}_i$ 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

$$\vec{v}_i \cdot \vec{v}_j \leq -\frac{1}{h-1}.$$

The *vector chromatic number* $\overline{\chi}(G)$ 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* [12]: choose a random vector $\vec{b}$, and retain all vertex vectors whose inner product with $\vec{b}$ 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** (*[9]*)**.** *Let $G = (V, E, w)$ be a weighted graph and $\ell, p$ 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 $\overline{\chi}(G_1) \leq l$.*

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

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 $\epsilon$ in time polynomial in $\log(1/\epsilon)$ and $n$ using the ellipsoid method [7] and Incomplete Choleski decomposition, as indicated in [12].

*Analysis*

We proceed to bound the weight and the independence number of the different (hyper)graphs computed by the algorithm, ending with the final solution $\hat{H}$. We first show that neither of these are reduced much in the subgraph $G_0$.

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

**Proof.** Recall that the algorithm defines the constants $a = \frac{1}{1-k/\log(k\bar{D})}$, $k = w(H)/\alpha$, and $c = \sqrt{\frac{ak-2}{ak} \ln(2k\bar{D})}$.

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 $G$ contains only a subset of the edges of $H$. Let $X = V(G) - V(G_0)$ be the high-degree vertices deleted to obtain $G_0$. Then,

$$\sum_{v\in X} w(v)d(v) \geq \sum_{v\in X} w(v)\cdot 2k\bar{D} = 2k\bar{D}w(X). \tag{1}$$

Since

$$\bar{D}\cdot w(V) = \sum_{v\in V} w(v)d(v) \geq \sum_{v\in X} w(v)d(v), \tag{2}$$

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

$$\alpha(G_0, w) \geq \alpha(G, w) - w(X) \geq \alpha(H, w) - w(X) \geq \frac{w(H)}{2k}. \quad \square$$

Observe that

$$\alpha(G_0, w) \geq \frac{w(H)}{2k} \geq \frac{w(G_0)}{2k} = \frac{w(G_0)}{2ak} + \frac{w(G_0)(a-1)}{2ak}.$$

Theorem 3.2 then ensures that a subgraph $G_1$ can be found with $w(G_1) \geq w(G_0)(a-1)/2ak$ and $\overline{\chi}(G_1) \leq 2ak$. A vector $2ak$-coloring of $G_1$ can then be found.

The main task is to bound the properties of the rounded subgraph $G_2$. Karger et al. [12] 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** ([12]). *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** ([16]). *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.** *The expected weight of $\hat{V}$ is bounded from below by*

$$\mathbb{E}[w(\hat{V})] = \Omega\left(\frac{w(G_1)}{\bar{D}^{1/2-1/k}\sqrt{k\log\bar{D}}}\right).$$

*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 $uv$ in $G_1$ 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 $w(E_1) = \sum_{vu\in E(G_1)}(w(v) + w(u))$ 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 $G_2$. The weight of vertices in $G_2$ with degree greater than $\sqrt{2k\bar{D}}$ is at most

$$\frac{\sum_{v_i\in V_2} w(v_i)d(v_i)}{\sqrt{2k\bar{D}}} = \frac{w(E_2)}{\sqrt{2k\bar{D}}}.$$

After deleting all such vertices from $G_2$, the expected weight of $\hat{V}$ is at least $\mathbb{E}[Z]$, where $Z$ denotes $w(V_2) - \frac{w(E_2)}{\sqrt{2k\bar{D}}}$. Our objective is to bound this quantity from below.

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 have that

$$\mathbb{E}[w(V_2)] = w(V_1)N(c). \tag{3}$$

Similarly, we bound $\mathbb{E}[w(E_2)]$ by

$$\mathbb{E}[w(E_2)] = w(E_1)N(c\tau) \leq 2k\bar{D}w(V_1)N(c\tau), \tag{4}$$

where in the last inequality we use the fact that the maximum degree of $G_1$ is by definition at most $2k\bar{D}$. Combining (3) and (4), we get that

$$\mathbb{E}[Z] = \mathbb{E}[w(V_2)] - \mathbb{E}\left[\frac{w(E_2)}{\sqrt{2k\bar{D}}}\right] \geq w(V_1)N(c) - \sqrt{2k\bar{D}}w(V_1)N(c\tau). \tag{5}$$

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,

$$N(c\tau) < \phi(c\tau)\frac{1}{c\tau} = \frac{(2k\bar{D})^{-1+1/ak}}{\sqrt{2\pi}\cdot\sqrt{\frac{2(ak-1)}{ak}\ln(2k\bar{D})}} \tag{6}$$

and

$$N(c) > \phi(c)\frac{1}{c}\left(1-\frac{1}{c^2}\right) = \frac{(2k\bar{D})^{-1/2+1/ak}}{\sqrt{2\pi}\cdot\sqrt{\frac{ak-2}{ak}\ln(2k\bar{D})}}\left(1-\frac{ak}{(ak-2)\ln(2k\bar{D})}\right). \tag{7}$$

Combining (5)–(7), we deduce that

$$\begin{aligned}\mathbb{E}[Z] &> w(V_1)\frac{(2k\bar{D})^{-1/2+1/ak}}{\sqrt{2\pi}\cdot\sqrt{\frac{ak-2}{ak}\ln(2k\bar{D})}}\left(1-\frac{ak}{(ak-2)\ln(2k\bar{D})}-\sqrt{\frac{ak-2}{2(ak-1)}}\right)\\ &= \Omega\left(\frac{w(V_1)}{(k\bar{D})^{1/2-1/ak}\sqrt{\log(k\bar{D})}}\right).\end{aligned} \tag{8}$$

Note that $\frac{1}{ak} = \frac{1}{k} - \frac{1}{\log(k\bar{D})}$, so $(k\bar{D})^{1/ak-1/2} = \frac{1}{2}(k\bar{D})^{1/k-1/2}$. Also, since the bounds we obtain are only interesting when $k = O(\log\bar{D})$, we have $\log(k\bar{D}) = \theta(\log\bar{D})$. Since $k^{1/k} = \theta(1)$ we can simplify (8) to get

$$\mathbb{E}[Z] = \Omega\left(\frac{w(V_1)}{\bar{D}^{1/2-1/k}\sqrt{k\log\bar{D}}}\right).$$

Finally, we can apply a derandomization technique from [15] 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 $\hat{H}$ from below in terms of the original hypergraph, using Theorem 3.2 and Lemma 3.3, by

$$w(G_1) \geq \frac{(a-1)w(G_0)}{2ak} = \frac{w(G_0)}{2\log(k\bar{D})} = \Omega\left(\frac{w(H)}{\log\bar{D}}\right).$$

Combined with Lemma 3.6, this gives the following result.

**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*

$$\Omega\left(\frac{w(H)}{\bar{D}^{1/2-1/k}\sqrt{k}(\log\bar{D})^{3/2}}\right),$$

*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 [18]. Given a hypergraph $H(V, E)$ with rank $r$, for any vertex $v \in V$ let $\bar{d}(v) = (d_1(v), \ldots, d_r(v))$ 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

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

and let $F(H) = \sum_{v\in V} f(\bar{d}(v))$. The GREEDY algorithm iteratively chooses a vertex $v \in V$ with $F(H\backslash v) \geq F(H)$ 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 [5] 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 [18] 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. Namely, each edge $(v_1, v_2, \ldots, v_t)$ (in some ordering of the vertices), is replaced by the edge $(v_1, v_2, v_3)$. 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 [18], GREEDY finds an independent set in a rank-3 hypergraph of size at least

$$\alpha(H') \geq n \sum_{j=0}^{d} \sum_{i=0}^{\sqrt{d}} \binom{d}{j}\binom{\sqrt{d}}{i} \frac{(-1)^{(j+i)}}{i+2j+1}. \tag{9}$$

By using the equality $\sum_k \binom{n}{k} \frac{(-1)^k}{x+k} = x^{-1}\binom{x+n}{n}^{-1}$ we can simplify (9) as:

$$\begin{aligned} \alpha(H') &\geq n \sum_{i=0}^{\sqrt{d}} (-1)^i \binom{\sqrt{d}}{i} \frac{1}{2} \left( \sum_{j=0}^{d} \binom{d}{j} \frac{(-1)^j}{j+(i+1)/2} \right) \\ &= \frac{n}{2(d+1)} \sum_{i=0}^{\sqrt{d}} (-1)^i \binom{\sqrt{d}}{i} \binom{(i+1)/2+d}{d+1}^{-1}. \end{aligned} \tag{10}$$

If we can show that for any value of $d$

$$F_d = \sum_{i=0}^{\sqrt{d}} (-1)^i \binom{\sqrt{d}}{i} \binom{(i+1)/2+d}{d+1}^{-1} \tag{11}$$

is lower bounded by $x\sqrt{d}$ for some $x > 0$, then, from (10) the GREEDY algorithm finds an independent set of size $\Omega(n/\sqrt{d})$.

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,

$$F_d = \sum_{i=0}^{\sqrt{d}} (-1)^i f_d(i). \tag{12}$$

We define

$$q_d(i) = \frac{(i+2)(i+4)\cdots(i+2d+2)}{(i+3)(i+5)\cdots(i+2d+1)} \tag{13}$$

for any $i \geq 0$. Using Stirling's approximation for the factorial function[1] 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

$$q_d(1) = \frac{(2d+3)!}{2^{2d+1}(d+1)!(d+1)!} = 4\sqrt{\frac{d}{\pi}}\left(1 + O\left(\frac{1}{d}\right)\right).$$

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

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 (13) we have that $\frac{f_d(i+1)}{f_d(i)} = \frac{\sqrt{d}-i}{q_d(i)} < 1$. From (11) and (12) it follows that $F_d > f_d(0) - f_d(1)$ and $f_d(0) = q_d(0)$, then

$$\begin{aligned} F_d &> f_d(0) - f_d(1) \\ &= f_d(0)\left(1 - \frac{\sqrt{d}}{q_d(0)}\right) \\ &= q_d(0) - \sqrt{d} \\ &= (\sqrt{\pi} - 1)\sqrt{d}\left(1 + O\left(\frac{1}{d}\right)\right). \end{aligned} \tag{14}$$

Thus, from (10), (11) and (14) the Greedy algorithm finds an independent set of size $\Omega(n/\sqrt{d})$. □

The bound on the performance ratio of GreedySDP then follows from Lemma 4.1 (with $d = 2k\bar{D} = 2k\bar{d}$), 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 $\bar{d}$, the* GreedySDP *algorithm finds an independent set of size* $\Omega\left(\frac{n}{k\bar{d}^{1-1/k}\log^{3/2}\bar{d}}\right)$.

Note that Greedy alone finds an independent set of size at least $n/(\bar{d}+1)$. So, if the maximum independent set is relatively large, or $O\left(\frac{n\log\log\bar{d}}{\log\bar{d}}\right)$, then it achieves a $O\left(\frac{\bar{d}\log\log\bar{d}}{\log\bar{d}}\right)$-approximation. On the other hand, if the maximum independent set in $H$ is at least $\frac{2n\log\log\bar{d}}{\log\bar{d}}$, i.e. $k \leq \frac{\log\bar{d}}{2\log\log\bar{d}}$, then GreedySDP finds a solution of size $\Omega(n/(k\bar{d}))$, for a $O(\bar{d})$-approximation. Therefore, we run both Greedy and GreedySDP and output the larger independent set found. We call this combined algorithm GreedySDP-MIS.

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

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

## 5. Randomized algorithm

The RandomIS algorithm extends the randomized version of Turán bound on graphs and was analyzed by Shachnai and Srinivasan in [17]. Given a hypergraph $H(V, E)$, the algorithm creates a random permutation $\pi$ of $V$ and adds a vertex $v$ to the independent set $I$, if there is no edge $e$ containing $v$ such that $v$ appears last in $\pi$ 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 $\pi$.

Shachnai and Srinivasan [17] 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 [5].

**Theorem 5.1** ([17], Theorem 2)**.** *For any $r \geq 2$ and any $r$-uniform hypergraph $H$,* RandomIS *finds an independent set of size at least*

$$\sum_{v\in V}\binom{d(v)+1/(r-1)}{d(v)}^{-1} = \Omega\left(\sum_{v\in V}\frac{w(v)}{(d(v))^{\frac{1}{r-1}}}\right).$$

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

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

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

**Theorem 5.2** *([17], Theorem 3).* *Given a weighted hypergraph $H'(V, E)$, the expected weight of the independent set produced by* RandomIS *is*

$$\Omega\left(\sum_{v\in V}\frac{w(v)}{a(v)^{1/b(v)}}f(v)\right),$$

*where $b(v) = \min_j(r_j(v) - 1)$.*

Shachnai and Srinivasan also show in [17] 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 $\Delta_2(H') \le \sqrt{2k\bar{D}}$ and maximum degree $\Delta(H') \le 2k\bar{D}$; and then uses RandomIS to find an independent set in $H'$.

**Theorem 5.3.** *Given a weighted hypergraph $H$ with average weighted degree $\bar{D}$, the* RandomSDP *algorithm finds an independent set of weight*

$$\Omega\left(\frac{w(H)}{k\bar{D}^{1-1/k}\log^{3/2}\bar{D}}\right).$$

**Proof.** We use that by the definitions of $a(v)$, $b(v)$ and $f(v)$, the independent set output by RandomIS is of weight at least

$$\sum_{v\in V}\frac{w(v)}{a(v)^{1/b(v)}}f(v) = \Omega\left(\sum_{v\in V}\frac{w(v)}{\sum_{i=2}^{r}d_i^{\frac{1}{i-1}}}\right) = \Omega\left(\frac{w(H')}{\Delta_2(H') + \sqrt{\Delta(H')}}\right) = \Omega\left(\frac{w(H')}{\sqrt{k\bar{D}}}\right).$$

The result then follows from Theorem 3.7. □

From Theorem 5.3 it follows that the RandomSDP algorithm approximates MIS within a factor of $O\left(\frac{\bar{D}}{\log\bar{D}}\right)$ if $\alpha(H,w) = \Omega\left(\frac{w(V)\log\log\bar{D}}{\log\bar{D}}\right)$, whereas RandomIS alone finds an approximation within a factor of $O\left(\frac{\bar{D}\log\log\bar{D}}{\log\bar{D}}\right)$ if $\alpha(H,w) = O\left(\frac{w(V)\log\log\bar{D}}{\log\bar{D}}\right)$. 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 $O\left(\frac{\bar{D}\log\log\bar{D}}{\log\bar{D}}\right)$.*

## 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 MIS in unweighted hypergraphs, matching the best known ratio for MIS 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.

## For further reading

[14].

## References

[1] N. Alon, N. Kahale, Approximating the independence number via the $\theta$-function, Math. Program. 80 (3) (1998) 253–264.
[2] P. Austrin, S. Khot, M. Safra, Inapproximability of vertex cover and independent set in bounded degree graphs, in: IEEE Conference on Computational Complexity, 2009 pp. 74–80.

[3] C. Bazgan, J. Monnot, V. Paschos, F. Serrière, On the differential approximation of MIN SET COVER, Theoret. Comput. Sci. 332 (2005) 497–513.
[4] V. Guruswami, A.K. Sinop, The complexity of finding independent sets in bounded degree (hyper)graphs of low chromatic number, SODA, 2011.
[5] Y. Caro, Z. Tuza, Improved lower bounds on $k$-independence, J. Graph Theory 15 (1991) 99–107.
[6] U. Feige, Approximating maximum clique by removing subgraphs, SIAM J. Discrete Math. 18 (2) (2005) 219–225.
[7] M. Grötschel, L. Lovász, A. Schrijver, The ellipsoid method and its consequences in combinatorial optimization, Combinatorica 1 (2) (1981) 169–197.
[8] M.M. Halldórsson, Approximations of independent sets in graphs, in: K. Jansen, J. Rolim (Eds.), Proc. APPROX, in: LNCS, vol. 1444, Springer, Berlin, 1998, p. 113.
[9] M.M. Halldórsson, Approximations of weighted independent set and hereditary subset problems, J. Graph Algorithms Appl. 4 (1) (2000) 1–16.
[10] M.M. Halldórsson, E. Losievskaja, Independent sets in bounded-degree hypergraphs, Discrete Appl. Math. 157 (2009) 1773–1786.
[11] E. Halperin, Improved approximation algorithms for the vertex cover problem in graphs and hypergraphs, SIAM J. Comput. 31 (5) (2002) 1608–1623.
[12] D. Karger, R. Motwani, M. Sudan, Approximate graph coloring by semidefinite programming, J. ACM 45 (2) (1998) 246–265.
[13] M. Krivelevich, R. Nathaniel, B. Sudakov, Approximating coloring and maximum independent sets in 3-uniform hypergraphs, J. Algorithms 41 (1) (2001) 99–113.
[14] L. Lovász, On Shannon capacity of a graph, IEEE Trans. Inform. Theory 25 (1) (1979) 1–7.
[15] S. Mahajan, H. Ramesh, Derandomizing semidefinite programming based approximation algorithms, SIAM J. Comput. 28 (5) (1999) 1641–1663.
[16] A. Rényi, Probability Theory, Elsevier, New York, 1970.
[17] H. Shachnai, A. Srinivasan, Finding large independent sets of hypergraphs in parallel, SIAM J. Disc. Math. 18 (3) (2005) 488–500.
[18] T. Thiele, A lower bound on the independence number of arbitrary hypergraphs, J. Graph Theory 32 (1999) 241–249.