Improved approximation algorithm for the Dense-3-Subhypergraph Problem

Amey Bhangale * Rajiv Gandhi † Guy Kortsarz ‡

Abstract

The study of Dense-3-Subhypergraph problem was initiated in Chlamtác et al. [7]. The input is a universe $U$ and collection $S$ of subsets of $U$, each of size 3, and a number $k$. The goal is to choose a set $W$ of $k$ elements from the universe, and maximize the number of sets, $S \in S$ so that $S \subseteq W$. The members in $U$ are called vertices and the sets of $S$ are called the hyperedges. This is the simplest extension into hyperedges of the case of sets of size 2 which is the well known Dense $k$-subgraph problem (see Kortsarz-Peleg [18]).

The best known approximation ratio for the Dense-3-Subhypergraph is $O(n^{0.69783..})$ by Chlamtác et al. [7]. We improve this ratio to $n^{0.61802..}$. More importantly, we give a new algorithm that approximates Dense-3-Subhypergraph within a ratio of $O(n/k)$ which improves the ratio of $O(n^2/k^2)$ of Chlamtác et al. [7].

We prove that under the log density conjecture (see Bhaskara et al. [5]) the ratio cannot be better than $\Omega(\sqrt{n})$ and demonstrate some cases in which this optimum can be attained.

1 Introduction

The Dense $k$-Subgraph (DkS) problem and its minimization version, called here the $p$-Min Dense subgraph (PMDS) problem are two well known problem with many applications. In the DkS problem we are given as input a graph $G = (V, E)$ and an integer $k$, and the goal is to find a subset $V' \subseteq V$ with $|V'| = k$ which maximizes the number of edges in the subgraph of $G$ induced by $V'$. In the minimization version, the Minimum Dense Subgraph (PMDS) problem, we are given a graph $G$ and a lower bound $p$ on the number of required edges. The goal is to find a least cardinality set $V' \subseteq V$ so that the subgraph induced by $V'$ contains at least $p$ edges. These problems have proved to be extremely useful. For example, a variant of DkS was recently used to obtain a new cryptographic system [3]. The same variant of the DkS problem was shown to be central in understanding financial

---

*Department of Computer Science, Rutgers New Brunswick Email: amey.bhangale@rutgers.edu
†Computer Science Department, Rutgers University, Camden, NJ, USA. Partially supported by NSF grants 1218620 and 1433220. Email: rajivg@camden.rutgers.edu
‡Computer Science Department, Rutgers University, Camden, NJ, USA. Partially supported by NSF grants 1218620 and 1540547. Email: guyk@crab.rutgers.edu
derivatives [4]. The best-known algorithms for many other problems involve using an algorithm for DkS or PMDS as a black box (e.g. [21, 15, 12]).

The first approximation ratio for DkS was $O(n^{2/5})$ [18] and was devised in 1993. Today, the best known ratio for the DkS problem is $O(n^{1/4+\epsilon})$ for arbitrarily small constant $\epsilon > 0$ [5], and the best known approximation for PMDS is $O(n^{3-2\sqrt{2}+\epsilon})$ for arbitrarily small constant $\epsilon > 0$ [8]. These problems are not known to be hard to approximate under the assumption $P \neq NP$.

The Dense-3-Subhypergraph problem admits as input a universe $U$, a collection $S$ of subsets of $U$ of size 3, and a number $k$. The goal is to choose a set $W$ of $k$ elements from the universe, and maximize the number of sets, $S \in S$ so that $S \subseteq W$. The first paper to study this problem gave the following approximation.

**Theorem 1.1.** [7] For every constant $\epsilon > 0$, there is an $O(n^{4(4-\sqrt{3})/13+\epsilon}) \leq O(n^{0.697831+\epsilon})$-approximation for the Dense-3-Subhypergraph problem.

In [7], the minimization of the extension of DkS to general hypergraphs is studied. This problem is called the Minimum $p$-union (MPU) problem. Given a set system $S$ over a universe $U$ and a number $k$, the goal is to find $k$ sets $S_1, S_2, \ldots, S_k$ from $S$ and minimize $|\bigcup_{i=1}^{k} S_i|$.

In all our problems, the members of the universe $U$ are called the vertices. The subsets of $U$ in $S$ are called the hyperedges.

**Theorem 1.2.** [9] The MPU problem admits an $m^{1/4+\epsilon}$ for any constant $\epsilon$ with $|S| = m$.

This is an improvement of the ratio $O(\sqrt{m})$ given by [7].

A special case of Dense-3-Subhypergraph and Minimum $p$-Union, the sets are all the neighborhoods of vertices in some graphs $G(U, E)$ (note that $G$ is a graph, not a hypergraph). More specifically, for every vertex $v \in U$, the induced graph $G_v(U, E_v)$ on the set of neighbors defined as $E_v = \{(u, w) \mid \{u, v, w\} \in S\}$ belongs to a special class. In [7], a polynomial time algorithm is given for the case when all the induced graphs on neighborhood sets are interval graphs.

**Theorem 1.3.** [7] Dense-$k$-Subhypergraph (DkSH) and Minimum $p$-Union (MPU) can be solved in polynomial time if all the induced graphs on neighborhood sets are interval graphs.

1.1 Our results

We prove the following results:

1. We give an $n^{0.61802}$-approximation for the Dense-3-Subhypergraph problem. Our improvement stems from a new technique that give a $\tilde{O}(n/k)$ ratio. This improves the ratio $O(n^2/k^2)$ of [7]. We present this result in Section 4 and 5.

2. On the other direction, in Section 3 we use the log density scheme to “predict” lower bounds. Under the so called log-density conjecture, (see [5], [9]), we prove that the best ratio we can
expect is $O(\sqrt{n})$.

3. Finally, in Section 6 we prove that without loss of generality we may assume some relations between the parameters. Let the average number of hyperdegree in the input graph be denoted by $D$ and $d^*$ be the average hyperdegree of the optimum.

We may assume without loss of generality that $D = \Theta(n^2/k^2)$ for a given instance without loss of generality. Thus, using $\hat{O}(n/k)$ approximation algorithm, the ratio equals $\hat{O}(\sqrt{D})$ and thus if $D = O(n)$ then we get $\hat{O}(\sqrt{n})$ ratio. We show that we may assume that $D = O(n\sqrt{n})$, or otherwise, the $\hat{O}(\sqrt{n})$ ratio follows. Also if $k \geq \sqrt{n}$ or $d^* = \Omega(k^2)$ or $d^* = O(k)$ we achieve $\hat{O}(\sqrt{n})$ ratio.

1.2 Related Work

The DkSH problem is the same as the Dense-3-Subhypergraph problem, except that the hyperedges have arbitrary size. Both DkSH and MPU problems were introduced by Applebaum [2] in the context of cryptography: Applebaum showed that if certain one way functions exist (or that certain pseudorandom generators exist) then DkSH is hard to approximate within $n^\epsilon$ for some constant $\epsilon > 0$. Based on this result, DkSH and MPU are used to prove hardness for other problems, such as the $k$-route cut problem [11]. To the best of our knowledge, the only two papers to deal with approximating these problem are [7] and [9]. In [7] it is shown that a ratio of $f$ for DkSH implies a ratio of $O(f \log p)$ for MPU.

In [17] it is shown that unless NP problems admit a subexponential randomized algorithm, the DkS problem admits no polynomial time approximation scheme (PTAS). In [13], the author presents an interesting conjecture about the possibility of detecting untypical random 3-SAT instances that have significantly more than 7/8 fraction of the clauses that can be simultaneously satisfied. The author assumes there is no polynomial time algorithm that for random SAT instances, if, say, 8/9 fraction of the clauses can be simultaneously satisfied, always says untypical, and for any other case the algorithm returns typical with high probability. This assumption implies that the DkS problem admits no PTAS. In [11], graph products were used to give the first non constant lower bound for the DkS problem, under the assumption of [13].

The following is due to Impagliazzo, Paturi and Zane, [16] and to the the sparsification lemma of [6], by Calabro, Impagliazzo and Paturi.

The **Exponential time hypothesis (ETH):** The SAT problem admits no $2^{o(n+m)}$ exact solution, with with $n$ the number of variables in the instance, and $m$ the number of clauses.

In a recent breakthrough [19], it is shown that under the ETH, the DkS problem admits no $n^{1/(\log \log n)^c}$ ratio for some constant $c$. Note that this implies the same hardness to the well known Label Cover problem (see [5]).
2 Preliminaries and Notation

The hypergraph is denoted by \( H = (U, \mathcal{S}) \). We use \( n = |U| \) and \( m = |\mathcal{S}| \) to denote the number of vertices and hyperedges respectively. The hyperdegree of a vertex in a hypergraph is the number of hyperedges which contain the vertex.

Given a subset \( V' \subseteq V \), the subhypergraph of \( H \) induced by \( V' \) is \( H[V'] = (V', \mathcal{S}_H) \) where \( \mathcal{S}_H = \{ S \in \mathcal{S} : S \subseteq V' \} \). Given a graph \( G = (V, E) \) and a vertex \( v \in V \), we use \( N(v) \) to denote the set of nodes adjacent to \( v \), and for a subset \( V' \subseteq V \) we let \( N(V') = \bigcup_{v \in V'} N(v) \). The average number of hyperdegree in the input graph is denoted by \( D \). We denote by \( \text{OPT} \) the optimal solution, and by \( d^* \) the average hyperdegree of the optimum.

3 Random graphs versus planted graphs

3.1 Random models versus planted model: the log density conjecture

In [5], the study of distinguishing random graphs from random graphs with a dense planted subgraph w.r.t. designing approximation algorithm for DkS problem was initiated. They define a quantity call log density which was crucial in their analysis.

**Definition 3.1.** The log density of a graph is the log to the base of the number of vertices, of the average degree of the graph.

The following conjecture appears explicitly in [9] and implicitly in [5]. Say that we have a graph \( G \) with log-density \( \alpha \). Further, the log density of the planted graph is \( \beta \) and \( \beta \leq \alpha - \epsilon \) for some arbitrarily small \( \epsilon \).

**Conjecture 3.2.** For \( k \leq \sqrt{n} \), in the above case we cannot distinguish in polynomial time between the case of Dense graph vs Random with planted subgraph problem.

Assumption 1: Assume that the planted graph has average hyperdegree that is larger by more than a \( \rho \) factor times than the average degree of \( G \). Then,

**Corollary 3.3.** Under Conjecture 3.2 and Assumption 1, the Dense-k-Subgraph problem admits no \( \rho \) approximation (for otherwise we can distinguish between the two cases).

The first (almost) tight success of the conjecture is given in [5] in which a ratio of \( O(n^{1/4+\epsilon}) \) is given for the DkS. This is complemented by a planted subgraph with smaller log density than \( G \), whose average hyperdegree is \( n^{1/4-\epsilon} \) higher than the one in \( G \). We (ignore the \( \epsilon \)) consider such an approximation to be tight under the Log Density conjecture.

In [9], a tight approximation of \( m^{1/4+\epsilon} \) is given for MPU, under a generalization of the log density conjecture. A third successful tight result under a (generalization of the) log density conjecture, is given for some cases of approximating "Projection games" [10]. The Projection game problem is also known as the Labelcover problem with the projection property (see for example [20]).
The Dense-3-Subhypergraph may be the closest problem to the Dense $k$-subgraph.

**Definition 3.4.** The hypergraph log density is log to the base of $n$ to the average hyperdegree of $H$.

With this definition of hypergraph log density, we now conjecture that similar phenomenon occurs in case of 3-uniform hypergraph. More specifically,

**Conjecture 3.5.** Let $k \leq \sqrt{n}$. Let $H$ has hypergraph log-density $\alpha$ and the planted hypergraph has hypergraph log density $\beta$, and $\beta \leq \alpha - \epsilon$ for some arbitrarily small $\epsilon > 0$. If the average hyperdegree of the planted graph is larger by a factor larger than $\rho$ than the average hyperdegree of $H$. Then the Dense-3-Subhypergraph problem admits no polynomial time $\rho$ approximation algorithm.

### 3.2 A $\sqrt{n}$ log density lower bound

Consider the following two hypergraphs.

1. $H_1$: For a set of $n$ vertices select each triplet as a hyperedge with probability $1/n$ and put $k = \sqrt{n}$. A simple calculation shows that the optimum number of hyperedges in every $k$ vertices subhypergraph is $\Theta(\sqrt{n})$ w.h.p.

2. $H_2$: Consider on the other hand planting a random subhypergraph of size $k$ in $H_1$. The set $U$ can be chosen at random. Every triplet of $U$ is set to be a hyperedge with probability slightly less than $1/k$, then the optimum will contains $p \cdot k^3$ hyperedges for $p$ slightly smaller than $1/k$. This gives average hyperdegree roughly $k$ for $k = \sqrt{n}$.

The average hyperdegree in the hypergraph $H_1$ is $O(1)$. Thus in $H_2$, the planted hypergraph has density about $\sqrt{n}$ larger than the entire hypergraph.

We now check the log density condition. In the hypergraph $H_1$, the average hyperdegree is very close to $n$ with high probability. This gives log-density of 1. The average hyperdegree in the planted hypergraph in $H_2$ is slightly less than $k = \sqrt{n}$ and so the log density is slightly less than 1. Hence, by the log density conjecture, a ratio better than $O(\sqrt{n})$ can not be found.

### 4 A $\tilde{O}(n/k)$ approximation for the Dense-3-Subhypergraph problem

In this section, we present our $\tilde{O}(n/k)$ approximation algorithm for Dense-3-Subhypergraph problem. We start with some preliminaries.
4.1 Preliminaries

We start by dividing $U$ at random into two equal sets $U_1$ and $U_2$. For a hyperedge $S = \{x_1, x_2, x_3\}$, we say that the base belongs to $U_1$ (respectively $U_2$) if exactly two of the vertices, say $x_1, x_2 \in U_1$. The third vertex $x_3$ is called the head of the hyperedge.

Definition 4.1. We say that an hyperedge is proper if it has a base in $U_1$ and a head in $U_2$.

The random partition makes sure that with high probability a constant fraction of hyperedges in the optimum are proper. The random choice is not required and is used for simplicity of the exposition. We can take an arbitrary disjoint partition $U_1, U_2$ into equal size sets, and then approximate the number of hyperedges with base in $U_1$, and the ones whose base is in $U_2$. Then we recursively call to find if most hyperedges are totally contained in $U_1$ or in $U_2$. The time required will be $T(n) = \text{poly}(n) + 2T(n/2)$ namely, a polynomial running time.

Consider some subset of $U_1$, say $U'$ of size $k$. Our goal is to check if the number of proper hyperedges of the optimum with base in $U'$ is large enough. The assumption $|U'| = k$ is for simplicity. Indeed, we are also going to choose $k$ vertices of $U_2$, as well. The number of vertices chosen is $2k$ by now, but taking $k$ vertices at random only decreases the number of hyperedges by a constant factor a thing that we may ignore. And this procedure can easily be derandomized with the method of conditional expectations.

Note that given $U' \subseteq U$, we can easily calculate the $k$ vertices $W \subseteq U_2$ so that $U' \cup W$ will have the largest number of proper hyperedges with base in $U'$. For every $w \in U_2$ separately, we can go over all edges $e = (u, v)$ with both endpoints in $U'$. The hyperdegree of a vertex $x_1 \in U_2$ in $U'$ is the number of different edges $(x_2, x_3) \in G(U')$, so that $(x_1, x_2, x_3)$ is an hyperedge.

Note that for $x_1, x_2 \in U_2$, so that $x_1 \neq x_2$ all the proper hyperedges with head $x_1$ are disjoint to all proper hyperedges with heads in $x_2$. Since these hyperedges are pairwise disjoint for every two heads, the best way to complete $U'$ to a dense subhypergraph is taking the $k$ vertices of $U_2$ with largest hyperdegree in $U'$. This motivates the following two definitions.

Definition 4.2. Given a set $U' \subseteq U_1$, let $c(U')$ be the set of $k$ vertices with highest hyperdegree in $U'$.

Definition 4.3. Given a set $U' \subseteq U_1$ and let $W \subseteq U_2$ be a subset of $U_2$. We denote by $h(U', W)$ be the number of hyperedges with base in $U'$, and head in $W$ (and base in $U'$).

Following definition will be useful in describing the algorithm concisely.

Definition 4.4. To nullify an hyperedge is to remove it from the list of the hyperedges. Note that we are not removing any vertices or edges, but just change the status of the nullified hyperedges, by taking these hyperedges out of $S$. 

6
We denote the optimum by $\text{OPT}$ and the average hyperdegree by $d^*$. We may guess the value $d^*$ of $\text{OPT}$ by going over all via binary search. Let $U'$ be the $k$ vertices of $U_1$ with largest hyperdegree. Given $U'$, we want to check if there are roughly $k \cdot (d^*k)/n$ hyperedges that belong to the optimum and also have base in $U'$. This is done by taking $c(U')$ and counting the number of hyperedges in this hypergraph. If $H(U', c(U')) \geq k \cdot (d^*k)/n$, we return $U' \cup c(U')$. Otherwise we nullify all proper hyperedges with base on $U'$. The main idea is that if only $k(d^*k)/n$ hyperedges of the optimum were lost. This number is “small” compared to the number of hyperedges nullified in the hypergraph. This implies that the hypergraph can not turn empty until few iterations because the optimum will not turn empty until few iteration unless we get at least $k \cdot (d^*k)/n$ hyperedges at some iteration. Thus there must be an iteration in which we get roughly $k \cdot (d^*k)/n$ hyperedges. This implies $\tilde{O}(n/k)$ ratio.

We present an iteration with a general guess $d^*_1$ for $d^*$. If $d^*_1 > d^*$ then the hypergraph may turn empty. However, it does not turn empty when we run it with $d^*$. Since we do not know $d^*$, we will return the largest value $X \geq d^*$ for which the hypergraph did not turn empty and we get the required number of hyperedges at some iteration. Thus, our ratio may only be better than $\tilde{O}(n/k)$ if $X >> d^*$.

### 4.3 The algorithm

The following algorithm (see Figure 1) uses binary search to guess $d^*$. It returns the hypergraph of the largest number for which the hypergraph did not turn empty.

After choosing at random $U_1, U_2$ we concentrate on proper hyperedges (edges with base in $U_1$). With a high probability, a constant fraction of the optimal hyperedges are proper. Abusing notation we still use $d^*$ for the average number of proper hyperedges in $\text{OPT}$. The average degree with respect to proper hyperedges is $\Theta(D)$ with high probability. For simplicity we still denote the average degree with respect to hyperedges by $D$. This deviates by constants from the real values of $d^*, D$ and thus, it is negligible for our context as we are aiming for $\omega(1)$ approximation algorithm. From now on, when we say hyperedges, we mean proper hyperedges hence we do not mention this assumption again.

### 4.4 Analysis

We now analyze the algorithm when $d^*_1 = d^*$.

**Claim 4.5.** Let $U'_i$ be the set $U'$ after $i - 1$ iterations. If the optimum has at least $k \cdot d^*/(3 \cdot n \cdot \ln n)$ hyperedges of $\text{OPT}$ with base in $U'_i$, we return a solution of value at least $k \cdot d^*/(3 \cdot n \cdot \ln n)$, getting
Algorithm Dense($H, k, d^*_1$)

1. Randomly partition $U$ into two sets $U_1$ and $U_2$.
   /* With high probability a constant fraction of the edges are proper */
2. $H' \leftarrow H$
3. while $H' \neq \emptyset$ do
   (a) Let $U'$ be the $k$ vertices of $U_1$ with largest hyperdegree.
      /* The hyperdegree is with respect to proper edges */
   (b) Compute $c(U') \subseteq U_2$.
   (c) If $U \cup c(U')$ contain at least $k \cdot (d^*_1 k)/(3 \cdot n \cdot \ln n)$ hyperedges return $U' \cup c(U')$.
   (d) Else nullify all proper hyperedges with base in $U'$.
4. If $H'$ turns empty return “Bad Guess: $d^*_1 > d^*$”.
5. Else return a random subset of $U' \cup c(U')$ of size $k$.

Figure 1: Algorithm for Dense-3-Subhypergraph

$a \tilde{O}(n/k) \cdot d^*_1$ hyperedges.

Proof. If the above holds, there exists a subset $U^*$ of $U_2 \cap \text{OPT}$ of size at most $k$ so that $h(U' \cup U^*) \geq k \cdot d^*_1/(3 \cdot n \cdot \ln n)$. By the definition of $c(U')$, $h(U' \cup c(U')) \geq h(U' \cup U^*) \geq k \cdot d^*_1/(3 \cdot n \cdot \ln n)$. The claim follows.

Claim 4.6. The number of iterations of Line 3 in Algorithm Dense at most $3 \cdot n/k \cdot \ln n$

Proof. Let $D_i$ be the average hyperdegree of vertices in $H'$ after $i - 1$ iterations. Let $D^*_i$ be the average hyperdegree of the set $U'_i$ of $k$ largest hyperdegree in $U'_i$. Since we take the $k$ highest hyperdegree vertices, their average hyperdegree is $D^*_i \geq D_i$. The algorithm does not stop unless all the hyperedges are nullified. The number of hyperedges not nullified drops from $n \cdot D_i$ to $n \cdot D_i - k \cdot D^*_i \leq n \cdot D_i (1 - k/n)$.

Recall that the initial average hyperdegree is $D$. Hence after $i = 3 \cdot k/(n \log n)$ iterations we get that $n \cdot D_i \leq n \cdot D \cdot (1 - k/n)^i$. For $i = 3 \cdot \ln n \cdot n/k$, we get an empty hypergraph because $D < n^2$ and so $n \cdot D < n^3$.

Claim 4.7. When we run the algorithm with the correct value of $d^*$, we always have a constant fraction $c \cdot (d^* k)$ for any $c < 1/e$ of optimal hyperedges that were not nullified, and so the optimum
can never turn empty

Proof. We prove that at least $c \cdot (d^* k)$ were not nullified with $c$ a constant slightly smaller than $1 - 1/e$. Let $k \cdot d^*_i$ be the number of remaining optimum hyperedges in OPT after round $i$. By the way the algorithm works, each time we iterate we get that $k \cdot d^*_{i+1} \geq k \cdot d^*_i \cdot \left(1 - \frac{k}{3 \cdot n \log n}\right)$. Thus after $3 \cdot \ln n \cdot n/k$ iterations,

$$k \cdot d^*_i \geq k \cdot d^* \cdot \left(1 - \frac{k}{3 \cdot n \cdot \ln n}\right)^{3 \cdot n \cdot \ln n/k} \geq c \cdot (d^* k),$$

where $c < 1/e$ as required. The last inequality follows from the fact that $(1 - 1/t)^t \sim 1/e$ for large $t$. \qed

Theorem 4.8. The algorithm Dense returns an $O(n/k)$ approximate solution.

Proof. We take the largest number for which the hypergraph does not turn empty. By Claim 4.7, for some $X \geq d^*$ the hypergraph does not turn empty. Indeed, in the worst case it will not turn empty when we run the algorithm with guess $d^*$, but will turn empty for guess $d^* + 1$. In any case, since the hypergraph did not turn empty with $X \geq d^*$, the number of hyperedges in the returned hypergraph is at least $k \cdot (Xk)/(3 \cdot \ln n \cdot n) \geq k \cdot (d^* k)/(3 \cdot \ln n \cdot n)$. Taking a random subhypergraph of $k$ vertices out of $U' \cup c(U')$ only looses a constant in the average hyperdegree. Thus, the ratio of $O(n/k)$ follows. \qed

Remark 4.9. Note that the approximation ratio of $k^2$ follows trivially from the fact that $d^* < k^2$ and we may return $k/3$ different hyperedges and get an average hyperdegree 1. This gives a ratio of $O(n^{2/3})$, if we take best of these two ($O(k^2)$ and $O(n/k)$ - approximation) algorithms. This already improves the ratio of $[7]$ without using hierarchy of LPs.

5 An improved ratio

We now improve this ratio using ideas of [7]. Let $\Delta_1$ be the average hyperdegree of the $k$ highest hyperdegree vertices. Define $\alpha$ and $\beta$ such that $n^\beta = \Delta_1 \cdot k/d^*$ and let $k = n^\alpha$.

Theorem 5.1. [7] There is an algorithm for Dense-3-Subhypergraph problem with approximation ratio at most $n^\gamma$ where

$$\gamma = \begin{cases} \alpha \cdot (2 - \alpha/\beta), & \text{if } \beta \leq 1 \\ \alpha \cdot (2 - \alpha), & \text{if } \beta \geq 1. \end{cases}$$

Remark 5.2. In [7] the ratio contains a multiplicative term $n^\epsilon$ for arbitrary small $\epsilon$ that we ignore.

We now prove our main theorem.

Theorem 5.3. The Dense-3-Subhypergraph problem admits a polynomial time algorithm with approximation ratio of $n^{0.61802\ldots}$. 

Proof. Since Algorithm Dense gives $O(n/k) = O(n^{1-\alpha})$ approximation ratio, if we take the best of Algorithm Dense and the one given by theorem 5.1 we get that the approximation ratio is at most $n^\gamma$ where

$$
\gamma = \begin{cases} 
\min\{\alpha \cdot (2 - \alpha/\beta), 1 - \alpha\}, & if \beta \leq 1 \\
\min\{\alpha \cdot (2 - \alpha), 1 - \alpha\}, & if \beta \geq 1.
\end{cases}
$$

Set $\mu := \alpha/\beta$. Note that as $\beta \leq 1$, $\alpha \leq \mu$. We get $\gamma \leq \min\{\alpha \cdot (2 - \mu), 1 - \alpha\}$ under the assumption that $\alpha \leq \mu$. If $\mu = \alpha + \epsilon$ the ratio does not increase but may decrease. Thus in the worst case, $\alpha = \mu$ (namely the worst case is for $\beta = 1$) and we get that

$$
\gamma \leq \min\{\alpha \cdot (2 - \alpha), 1 - \alpha\}.
$$

This ratio holds true for $\beta \leq 1$ and $\beta \geq 1$. This ratio is maximized for $\alpha = 0.38196601125$ which gives $\gamma \leq 0.61802..$ and hence the approximation guarantee follows. Making $\alpha$ slightly smaller, by changing the last digit to 4 increases $1 - \alpha$ and takes care of the fact that we ignored the $n^\epsilon$ term. This also takes care of any polylogarithmic term. The above ratio follows.

6 The ratio w.r.t. some parameters

6.1 Parameters that imply an Optimal ratio

Clearly, if $k \geq \sqrt{n}$ we get an optimal ratio (using $O(n/k)$ approximation) under the log density conjecture, because the log density conjecture predicts that we can not get better than $\sqrt{n}$ ratio. A second case in which we can get the ratio projected by the log density conjecture is:

Claim 6.1. If $d^* = \Omega(k^2)$ the problem admits a $n^{1/2+\epsilon}$ for any constant $\epsilon > 0$ and is almost optimal under the log density conjecture.

Proof. If $d^* = \Omega(k^2)$ we get that there exists a graph with at most $k$ vertices and $\Omega(k)$ average hyperdegree. Thus, there exists a vertex $v$ such that the graph $G_v$ induced by the hyperedges containing $v$ has a dense $k$ subgraph of average degree $\Omega(k)$. In Section 5.1 of [14], an $O(n^\epsilon)$ ratio is give for this case for any $\epsilon > 0$. Thus, using that algorithm we can get a dense $k+1$ subhypergraph (the vertex $v$ with the solution returned by running the algorithm from [14] on $G_v$) with $\Omega(k^2/n^\epsilon)$ hyperedges. This implies a ratio of $k \cdot n^\epsilon$ for the Dense-3-Subhypergraph problem. Combined with the $\tilde{O}(n/k)$ (taking the best of these two) this implies a $n^{1/2+\epsilon}$ ratio.

A trivial statement is that if $D = \Omega(n^2)$ the problem admits an $O(1)$ ratio (by just choosing $k$ vertices at random). We show a non trivial case for which a ratio of $\tilde{O}(\sqrt{n})$ ratio applies.
Claim 6.2. If \( D = \Omega(n\sqrt{n}) \) there exists a polynomial time algorithm for Dense-3-Subhypergraph with ratio \( \tilde{O}(\sqrt{n}) \).

Proof. We first prove that without loss of generality we may assume that \( D = \Theta(n^2/k^2) \). In [5] it is shown (in the appendix) that for the DkS problem, we may assume that \( D \leq n/k \). An identical proof with slightly different parameters gives that we may assume that \( D \leq n^2/k^2 \). The above inequality implies that if \( D = \Omega(n\sqrt{n}) \) then \( k^2 = \tilde{O}(\sqrt{n}) \), and using a ratio of \( k^2 \), the claim follows. \( \square \)

Claim 6.3. With respect to getting any poly\((n)\) approximation for Dense-3-Subhypergraph, we can assume without loss of generality \( D = \Theta(n^2/k^2) \).

Proof. If \( D > n^2/k^2 \), the the argument similar to one in [5], we can make \( D \sim n^2/k^2 \). For the other direction, if \( D << n^2/k^2 \), we add fake hyperedges. We later show that the number of fake hyperedges that belong to any subset of \( k \) is at most \( k \cdot \log^2 n \). This is a negligible number. If the number of hyperedges in the set we return is \( O(k \cdot \log^2 n) \), then with respect to polynomial algorithm nothing changes. Indeed returning an average degree 1 or average degree \( \log^2 n \) is the same when the ratio is polynomial. To add fake hyperedges put every hyperedge (among the \( \binom{n}{3} \) potential hyperedges) into the graph independently with probability \( 1/k^2 \). The expected number of hyperedges added is \( \binom{n}{3} \cdot 1/k^2 = \Theta(n^3/k^2) \), Thus using Chernoff bound, the average hyperdegree is \( \Theta(n^2/k^2) \) with very high probability as required.

We now bound the maximum average degree over all sets of size \( k \) with respect to fake edges. Consider any set \( U \) of \( k \) vertices. The expected number of fake hyperedges added to \( U \) is \( \Theta(k^3/k^2) = \Theta(k) \), which makes the average hyperdegree \( c \) for some constant \( c \). We bound the probability that the number of fake hyperedges added is \( \Omega(\log^2 n \cdot k) \). Thus using Chernoff bound, that the probability for that event is at most

\[
\Pr [ \text{number of fake hyperedges in } U \geq \Omega(\log^2 n \cdot k) ] \leq e^{-\Theta(k \ln^2 n)}.
\]

The number of subsets of size \( k \) is at most

\[
\binom{n}{k} \leq \left( \frac{e \cdot n}{k} \right)^k.
\]

By the union bound it follows that with high probability that the average of fake hyperedges for any set of \( k \) vertices is at most \( \Theta(\log^2 n) \) which is negligible (if \( d^* < \sqrt{n} \) we may return \( k/3 \) different hyperedges and get ratio \( \sqrt{n} \)). Hence it is possible to approximate the problem on the new graph, and after a set \( W \) of \( k \) vertices is output, remove all fake hyperedges within \( W \). This is significant only if the average hyperdegree of \( W \) is \( o(\log^2 n) \). But in that case it means that the set \( W \) output has negligible number of hyper edges to begin with and can not be used to get a better ratio. \( \square \)
If we assume $D = O(n)$ then using the above claim and $\tilde{O}(n/k)$ approximation ratio, we get the following corollary.

**Corollary 6.4.** If $D = O(n)$, we get $\tilde{O}(\sqrt{n})$ approximation ratio for Dense-3-subhypergraph problem.

### 6.2 The state of the art: worst parameters

The worst case for our algorithm is $k = n^{0.38196}$. As $\tilde{O}(n/k) = \sqrt{D}$, and any change in $\alpha$ gives a better ratio, the worst case for $D$ is $D = n^{1.23608}$. As we can also get a $d^*$ ratio, the worse choice of $d^*$ is probably $d^* = n^{1-\alpha} = n^{0.61802}$. In addition, it seems that the hardest case is if the graph is regular (or almost regular).

**Theorem 6.5.** Let $\Delta_1$ be the average hyperdegree of the $k$ largest vertices in $G$. Then the problem admits a solution with average degree $\Delta_1 \cdot k^2/n^2$

If, $\Delta_1 = \Theta(D)$ namely the graph is close to regular, Theorem 6.5 gives a trivial ratio. On the other hand if $\Delta_1 >> D$, holds for several iterations the $\tilde{O}(n/k)$ ratio can be improved because the number of hyperedges nullified is larger than $k \cdot D$. Thus it may be that the case of regular graph (in which $\Delta_1 = D$) is the hardest case.

### 7 An open problem

We note that even if all hyperedges are of size 3 the ratio of $m_{\text{avg}}$ remains $m^{1/4+\epsilon}$, a thing that seems counter intuitive and also is not tight in the log-density model. An interesting open problem, is to get a better ratio than $m^{1/4+\epsilon}$ for constant size hyperedges.

**Acknowledgment**

We would like to than Eden Chlamtác for discussion on the log-density scheme and Uriel Feige and Rohit Khandekar for helpful discussion.

**References**

[1] N. Alon, S. Arora, R. Manokaran, D. Moshkovitz, and O. Weinstein. Inapproximability of densest k-subgraph from average case hardness. Manuscript, 2016.

[2] B. Applebaum. Pseudorandom generators with long stretch and low locality from random local one-way functions. *SIAM Journal on Computing*, 42(5):2008–2037, 2013.
[3] B. Applebaum, B. Barak, and A. Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the Forty-second ACM Symposium on Theory of Computing*, STOC ’10, pages 171–180, 2010.

[4] S. Arora, B. Barak, M. Brunnermeier, and R. Ge. Complicational complexity and information asymmetry in financial products. Submitted, 2016.

[5] A. Bhaskara, M. Charikar, E. Chlamtac, U. Feige, and A. Vijayaraghavan. Detecting high log-densities: an $O(n^{1/4})$ approximation for densest $k$-subgraph. In *Proceedings of the 42nd ACM Symposium on Theory of Computing*, STOC 2010, Cambridge, Massachusetts, USA, 5-8 June 2010, pages 201–210, 2010.

[6] C. Calabro, R. Impagliazzo, and R. Paturi. A duality between clause width and clause density for sat. In *CCC*, pages 252–260, 2006.

[7] E. Chlamtác, M. Dinitz, C. Konrad, G. Kortsarz, and G. Rabanca. The densest $k$-subhypergraph problem. In *APPROX*, pages 6:1–6:19, 2016.

[8] E. Chlamtac, M. Dinitz, and R. Krauthgamer. Everywhere-sparse spanners via dense subgraphs. In *53rd Annual IEEE Symposium on Foundations of Computer Science*, FOCS 2012, New Brunswick, NJ, USA, October 20-23, 2012, pages 758–767, 2012.

[9] E. Chlamtác, M. Dinitz, and Y. Makarychev. Minimizing the union: Tight approximations for small set bipartite vertex expansion. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19, pages 881–899, 2017.

[10] E. Chlamtác, P. Manurangsi, D. Moshkovitz, and A. Vijayaraghavan. Approximation algorithms for label cover and the log-density threshold. In *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 2017, Barcelona, Spain, Hotel Porta Fira, January 16-19, pages 900–919, 2017.

[11] J. Chuzhoy, Y. Makarychev, A. Vijayaraghavan, and Y. Zhou. Approximation algorithms and hardness of the k-route cut problem. *ACM Trans. Algorithms*, 12(1):2:1–2:40, Dec. 2015.

[12] M. Dinitz, G. Kortsarz, and Z. Nutov. Improved Approximation Algorithm for Steiner k-Forest with Nearly Uniform Weights. In *APPROX*, volume 28, pages 115–127, 2014.

[13] U. Feige. Relations between average case complexity and approximation complexity. In *STOC*, 2002.

[14] U. Feige, G. Kortsarz, and D. Peleg. The dense $k$-subgraph problem. *Algorithmica*, 29(3):410–421, 2001.
[15] A. Gupta, M. Hajiaghayi, V. Nagarajan, and R. Ravi. Dial a ride from k-forest. \textit{TALG}, 6(2):41:1–41:21, 2010.

[16] R. Impagliazzo, R. Paturi, and F. Zane. Which problems have strongly exponential complexity? In \textit{FOCS}, pages 653–663, 1998.

[17] S. Khot. Ruling out ptas for graph min-bisection, dense k-subgraph, and bipartite clique. \textit{SIAM Journal on Computing}, 36(4):1025–1071, 2006.

[18] G. Kortsarz and D. Peleg. On choosing a dense subgraph. In \textit{34th Annual Symposium on Foundations of Computer Science, Palo Alto, California, USA, 3-5 November 1993}, pages 692–701, 1993.

[19] P. Manurangsi. Almost-polynomial ratio eth-hardness of approximating densest k-subgraph. In \textit{STOC}, page to appear, 2017.

[20] P. Manurangsi and D. Moshkovitz. Improved approximation algorithms for projection games. \textit{Algorithmica}, 77(2):555–594, 2017.

[21] Z. Nutov. Approximating steiner networks with node-weights. \textit{SIAM Journal on Computing}, 39(7):3001–3022, 2010.