Approximate coloring of uniform hypergraphs *

Michael Krivelevich †   Benny Sudakov ‡

Abstract

We consider an algorithmic problem of coloring r-uniform hypergraphs. The problem of finding the exact value of the chromatic number of a hypergraph is known to be NP-hard, so we discuss approximate solutions to it. Using a simple construction and known results on hardness of graph coloring, we show that for any $r \geq 3$ it is impossible to approximate in polynomial time the chromatic number of r-uniform hypergraphs on n vertices within a factor $n^{1-\epsilon}$ for any $\epsilon > 0$, unless $NP \subseteq ZPP$. On the positive side, improving a result of Hofmeister and Lefmann, we present an approximation algorithm for coloring r-uniform hypergraphs on n vertices, whose performance ratio is $O(n(\log\log n)^2/(\log n)^2)$.

1 Introduction

A *hypergraph* H is an ordered pair $H = (V, E)$, where V is a finite nonempty set (the set of *vertices*) and E is a collection of distinct nonempty subsets of V (the set of *edges*). H has *dimension* r if all edges have at most r vertices. If all edges have size exactly r, H is called r*-uniform*. Thus, a 2-uniform hypergraph is just a graph. A set $U \subseteq V(H)$ is called *independent* if U spans no edges of H. The maximal size of an independent set in H is called the *independence number* of H and is denoted by $\alpha(H)$. A k*-coloring* of H is a mapping $c : V(H) \to \{1, \ldots, k\}$ such that no edge of H (besides singletons) has all vertices of the same color. Equivalently, a k-coloring of H is a partition of the vertex set $V(H)$ into k independent sets. The *chromatic number* of H, denoted by $\chi(H)$ is the minimal k, for which H admits a k-coloring.

In this paper we consider an algorithmic problem of coloring r-uniform hypergraphs, for given and fixed value of $r \geq 2$. The special case $r = 2$ (i.e. the case of graphs) is relatively well studied and many results have been obtained in both positive (that is, good approximation algorithms, see e.g. [16], [26], [3], [13], [4], [17], [5]) and negative (that is, by showing the hardness of approximating

*A preliminary version of these results appeared in Proceedings of the Sixth Annual European Symposium on Algorithms (ESA'98), Lecture Notes in Computer Science 1461, Springer Verlag (1998), 477-489.

†DIMACS Center, Rutgers University, Piscataway, NJ 08854, USA. Research supported by an IAS/DIMACS Postdoctoral Fellowship. Current affiliation: Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv 69978, Israel. E-mail: krivelev@post.tau.ac.il.

‡Department of Mathematics, Raymond and Beverly Sackler Faculty of Exact Sciences, Tel Aviv University, Tel Aviv, Israel. Current affiliation: Department of Mathematics, Princeton University, Princeton, NJ 08544, USA. Email address: bsudakov@math.princeton.edu.

the chromatic number under some natural complexity assumptions, see e.g. [19], [11]) directions. We will briefly survey these developments in the subsequent sections of the paper. However, much less is known about the general case. Lovász [24] showed that it is NP-hard to determine whether a 3-uniform hypergraph is 2-colorable. Additional results on complexity of hypergraph coloring were obtained in [25], [8], [7]. These hardness results give rise to attempts of developing algorithms for *approximate* uniform hypergraph coloring, aiming to use a small but possibly non-optimal number of colors. The first non-trivial case of approximately coloring 2-colorable hypergraphs has been considered in papers of Chen and Frieze [9] and of Kelsen, Mahajan and Ramesh ([18], a journal version appeared in [1]). Both papers arrived independently to practically identical results. They presented an algorithm for coloring a 2-colorable r-uniform hypergraph in $O(n^{1-1/r})$ colors, using an idea closely related to the basic idea of Wigderson's coloring algorithm [26]. Another result of the above mentioned two papers is an algorithm for coloring 3-uniform 2-colorable hypergraphs in $\tilde{O}(n^{2/9})$ colors. The latter algorithm exploits the semidefinite programming approach, much in the spirit of the Karger-Motwani-Sudan coloring algorithm [17]. Not much is known about general approximate coloring algorithms for r-uniform hypergraphs (that is, when the chromatic number of a hypergraph is not given in advance). Recently Hofmeister and Lefmann [15] presented an approximation algorithm for this problem with performance ratio $O(n/(\log^{(r-1)} n)^2)$, where $\log^{(r)} n$ denotes the r-fold iterated logarithm.

This paper is aimed at trying to fill a gap between the special case of graphs ($r = 2$) and the case of a general r. We present results in both negative and positive directions. In Section 2 we describe a construction which enables to derive immediately results on hardness of approximating the chromatic number of r-uniform hypergraphs for any $r \geq 3$ from the corresponding graph results. Thus we get that unless $NP \subseteq ZPP$, for any fixed $r \geq 3$, it is impossible to approximate the chromatic number of r-uniform hypergraphs on n vertices in polynomial time within a factor of $n^{1-\epsilon}$, for any $\epsilon > 0$. It should be noted that Hofmeister and Lefmann obtained independently the same hardness result in [15].

In Section 3 we present an approximation algorithm for coloring r-uniform hypergraphs on n vertices, whose performance guarantee is $O(n(\log\log n)^2/(\log n)^2)$, thus matching the approximation ratio of Wigderson's algorithm [26]. This algorithm is quite similar in a spirit to the algorithm of Wigderson, though technically somewhat more complicated. Final Section 4 is devoted to concluding remarks.

All logarithms are natural unless written explicitly otherwise.

Remark. In the conference version of this paper [23] we also presented an algorithm for coloring 3-uniform 2-colorable hypergraphs on n vertices in $\tilde{O}(n^{9/41})$ colors, thus improving previous results of Chen and Frieze and of Kelsen, Mahajan and Ramesh. We omit this algorithm in the current journal version, since the rather slow refereeing process allowed us enough time to improve this result. Indeed, together with Nathaniel in [22] we obtained an algorithm for coloring 3-uniform 2-colorable hypergraphs on n vertices in $\tilde{O}(n^{1/5})$ colors, and we refer the interested reader to this new paper.

2 Hardness of approximation

Several results on hardness of calculating exactly the chromatic number of r-uniform hypergraphs have been known previously. Lovász [24] showed that it is NP-complete to decide whether a given 3-uniform hypergraph H is 2-colorable. Phelps and Rödl proved in [25] that it is NP-complete to decide k-colorability of r-uniform hypergraphs for all $k, r \geq 3$, even when restricted to linear hypergraphs. Brown and Corneil [8] presented a polynomial transformation from k-chromatic graphs to k-chromatic r-uniform hypergraphs. Finally, Brown showed in [7] that, unless $P = NP$, it is impossible to decide in polynomial time 2-colorability of r-uniform hypergraphs for any $r \geq 3$.

However, until recently, there has been no result showing that it is also hard to *approximate* the chromatic number of r-uniform hypergraphs, where $r \geq 3$. For the graph case ($r = 2$), Feige and Kilian showed in [11], using the result of Håstad [14], that if NP does not have efficient randomized algorithms, then there is no polynomial time algorithm for approximating the chromatic number of an n vertex graph within a factor of $n^{1-\epsilon}$, for any fixed $\epsilon > 0$.

In this section we present a construction for reducing the approximate graph coloring problem to approximate coloring of r-uniform hypergraphs, for any $r \geq 3$. Using this construction and the above mentioned result by Feige and Kilian we will be able to deduce hardness results in the hypergraph case. As we learned after having written a conference version of this paper [23], Hofmeister and Lefmann presented in [15] an essentially identical construction.

Let $r \geq 3$ be a fixed uniformity number. Suppose we are given a graph $G = (V, E)$ on $|V| = n \geq r$ vertices with chromatic number $\chi(G) = k$. Define an r-uniform hypergraph $H = (V, F)$ in the following way. The vertex set of H is identical to that of G. For every edge $e \in E$ and for every $(r-2)$-subset $V_0 \subseteq V \setminus e$ we include the edge $e \cup V_0$ in the edge set F of H. If $F(H)$ contains multiple edges, we leave only one copy of each edge. The obtained hypergraph H is r-uniform on n vertices. Now we claim that $k/(r-1) \leq \chi(H) \leq k$. Indeed, a k-coloring of G is also a k-coloring of H, implying the upper bound on $\chi(H)$. To prove the lower bound, let $f : V \to \{1, \ldots, k'\}$ be a k'-coloring of H. Let G_0 be a subgraph of G, whose vertex set is V and whose edge set is composed of all these edges of G that are monochromatic under f. It is easy to see that the degree of every vertex $v \in V$ in G_0 is at most $r-2$ (otherwise the union of the edges of G_0 incident with v would form a monochromatic edge in H). Thus G_0 is $(r-1)$-colorable. We infer that the edge set $E(G)$ of G can be partitioned into two subsets $E(G) \setminus E(G_0)$ and $E(G_0)$ such that the first subset forms a k'-colorable graph, while the second one is $(r-1)$-colorable. Then G is $k'(r-1)$-colorable, as we can label each vertex by a pair whose first coordinate is its color in a k'-coloration of the first subgraph, and the second coordinate comes from an $(r-1)$-coloration of the second subgraph. Therefore G and H as defined above have the same number of vertices, and their chromatic numbers have the same order. Applying now the result of Feige and Kilian [11], we get the following theorem.

Theorem 2.1 *Let $r \geq 3$ be fixed. If $NP \not\subseteq ZPP$, it is impossible to approximate the chromatic number of r-uniform hypergraphs on n vertices within a factor of $n^{1-\epsilon}$ for any fixed $\epsilon > 0$ in time polynomial in n.*

Remark. Several quite recent results [12], [20], [10], [21] show that it is hard to color r-uniform k-colorable hypergraphs in few colors, for various values of k and r. Specific statements can be found in the corresponding papers.

3 A general approximation algorithm

In this section we present an approximation algorithm for the problem of coloring r-uniform hypergraphs, for a general $r \geq 3$. Throughout the section the uniformity parameter r is assumed to be fixed.

Let us start with describing briefly the history and state of the art of the corresponding graph problem ($r = 2$), that is, of the problem of approximate graph coloring. As we have already mentioned in the introduction, this question is relatively well-studied. The first result on approximate graph coloring belongs to Johnson [16], who in 1974 proposed an algorithm with approximation ratio of order $n/\log n$, where n denotes the number of vertices in a graph G. The next step was taken by Wigderson [26], whose algorithm achieves approximation ratio $O(n(\log\log n)^2/(\log n)^2)$. The main idea of Wigderson's algorithm was quite simple: if a graph is k-colorable then the neighborhood $N(v)$ of any vertex $v \in V(G)$ is $(k-1)$-colorable, thus opening a way for recursion. Berger and Rompel [3] further improved Johnson's result by presenting an algorithm whose approximation ratio is better than that of Wigderson's algorithm by a factor of $\log n/\log\log n$. They utilized the fact that if G is k-colorable then one can find efficiently a subset S of a largest color class which has size $|S| \geq \log_k n$ and neighborhood $N(S)$ of size at most $n(1-1/k)$. Repeatedly finding such S and deleting it and its neighborhood leads to finding an independent set of size $(\log_k n)^2$. Finally, Halldórsson [13] came up with an approximation algorithm that uses at most $\chi(G)n(\log\log n)^2/(\log n)^3$ colors, currently best known result. His contribution is based on Ramsey-type arguments for finding a large independent set from his paper with Boppana [6]. Both papers [3] and [13] proceed by repeatedly finding a large independent set, coloring it by a fresh color and discharging it - quite a common approach in graph coloring algorithms. We will also adopt this strategy. It is worth noting here than one cannot hope for a major breakthrough in this algorithmic question due to the hardness results mentioned in Section 2.

Unfortunately, most of the ideas of the above discussed papers do not seem to be applicable smoothly to the hypergraph case (i.e., when $r \geq 3$). It is not clear how to define a notion of the neighborhood of a subset in order to apply the Berger-Rompel approach. Also, bounds on the hypergraph Ramsey numbers are too weak to lead to algorithmic applications in the spirit of [6], [13]. A Ramsey-based approach has been applied by Hofmeister and Lefmann in [15] to derive an algorithm with performance ratio $O(n/(\log^{(r-1)} n)^2)$ (here $\log^{(r)} n$ denotes the r-fold iterated logarithm), which becomes much weaker than the above cited graph algorithms as r grows. However, something from the graph case can still be rescued. Both papers [9] and [18], dealing with the case of 2-colorable hypergraphs, noticed that the main idea behind Wigderson's algorithm is still usable for the hypergraph case. Let us describe now the main instrument of these papers, playing a key role in our arguments as well. For a hypergraph $H = (V, E)$ and a subset of vertices $S \subseteq V$, let $N(S) = \{v \in V : S \cup \{v\} \in E\}$. The following procedure is used in both papers [9] and [18].

Procedure *Reduce*(H, S)
Input: A hypergraph $H = (V, E)$ and a vertex subset $S \subseteq V$.
Output: A hypergraph $H' = (V, E')$.

1. Delete from E the set of edges $\{S \cup \{v\} : v \in N(S)\}$;

2. Add to E an edge S, denote the resulting hypergraph by H'.

Proposition 3.1 *Let $H' = Reduce(H, S)$.*

1. *if U is an independent set in H', then U is independent in H;*
2. *If H is k-colorable and the induced subhypergraph $H[N(S)]$ is not $(k-1)$-colorable, then H' is k-colorable.*

Proof. The first part of the proposition is obvious. To prove the second part, fix a k-coloring $c : V(H) \to \{1, \ldots, k\}$ of H. If all vertices of S get the same color under c, say, they are all colored in color 1, then this color is not used in coloring $N(S)$, thus implying that the hypergraph $H[N(S)]$ is $(k-1)$-colorable and contradicting our assumption. Therefore S is not monochromatic in c, showing that c is a proper k-coloring of H' as well. $\Box$

Note that the above proposition replaces edges of H by an edge of smaller size. Therefore, in order to apply it we need to widen our initial task and instead of developing an algorithm for coloring r-uniform hypergraphs to present an algorithm for hypergraphs of dimension r. Based on Prop. 3.1, we can use a recursion on k for coloring k-colorable hypergraphs of dimension r. Indeed, if for some S the subset $N(S)$ is relatively large and $H[N(S)]$ is $(k-1)$-colorable, then applying recursion we can find a relatively large independent subset of $N(S)$. If $H[N(S)]$ is not $(k-1)$-colorable, we can use procedure *Reduce*(H, S) in order to reduce the total number of edges. Finally, when the hypergraph is relatively sparse, a large independent set can be found based on the following proposition.

Proposition 3.2 *Let $H = (V, E)$ be a hypergraph of dimension $r \geq 2$ on n vertices without singletons. If every subset $S \subset V$ of size $1 \leq |S| \leq r-1$ has a neighborhood $N(S)$ of size $|N(S)| \leq t$, then H contains an independent set U of size $|U| \geq \frac{1}{4}(n/t)^{1/(r-1)}$, which can be found in time polynomial in n.*

Proof. For every $2 \leq i \leq r$, Let E_i be the set of all edges of size i in H. Then $E = \bigcup_{i=2}^{r} E_i$. By the assumptions of the proposition we have

$$|E_i| \leq \frac{\binom{n}{i-1} t}{\binom{i}{i-1}} \leq n^{i-1} t \ .$$

Choose a random subset V_0 of V by taking each $v \in V$ into V_0 independently and with probability $p_0 \geq 1/n$, where the exact value of p_0 will be chosen later. Define random variables X, Y by letting X be the number of vertices in V_0 and letting Y be the number of edges spanned by V_0. Then

$$\begin{aligned} E[X] &= np_0 \ , \\ E[Y] &= \sum_{i=2}^{r} |E_i| p_0^i \leq \sum_{i=2}^{r} n^{i-1} t p_0^i \leq (r-1) n^{r-1} p_0^r t \ . \end{aligned}$$

Now we choose p_0 so that $E[X] \geq 2E[Y]$. For example, we can take $p_0 = \frac{1}{2}(n^{r-2}t)^{-1/(r-1)}$. Then by linearity of expectation there exists a set V_0, for which $X - Y \geq \frac{1}{4}(n/t)^{1/(r-1)}$. Fix such a set V_0 and for every edge e spanned by V_0 delete from V_0 an arbitrary vertex of e. We get an independent set U of size $|U| \geq X - Y \geq \frac{1}{4}(n/t)^{1/(r-1)}$.

The above described randomized algorithm can be easily derandomized using standard derandomization techniques (see, e.g., [2], Ch. 15). $\square$

We denote the algorithm described in Proposition 3.2 by $I(H, t)$. Here are its formal specifications.

Algorithm $I(H, t)$
Input: An integer t and a hypergraph $H = (V, E)$ of dimension r on n vertices, in which every $S \subset V$ of size $1 \leq |S| \leq r-1$ satisfies $|N(S)| \leq t$.
Output: An independent set U of H of size $|U| = \frac{1}{4}(n/t)^{1/(r-1)}$.

Now we are ready to give a formal description of a recursive algorithm for finding a large independent set in k-colorable hypergraphs of dimension r. Define two functions:

$$\begin{aligned} g_k(n) &= \frac{1}{4} n^{\frac{1}{(r-1)(k-1)+1}} \ , \\ f_k(n) &= n^{1 - \frac{r-1}{(r-1)(k-1)+1}} \ . \end{aligned}$$

One can easily check that g and f satisfy

$$g_{k-1}(f_k(n)) = g_k(n), \quad \frac{1}{4}\left(\frac{n}{f_k(n)}\right)^{\frac{1}{r-1}} = g_k(n) \ .$$

Algorithm $A(H, k)$
Input: An integer $k \geq 1$ and a hypergraph $H = (V, E)$ of dimension r.
Output: A subset U of V.

1. $n = |V(H)|$;
2. **if** $k = 1$ take U to be an arbitrary subset of V of size $|U| = g_k(n)$ and **return**(U);

3. **if** $k \geq 2$ **then**

4. **while** there exists a subset $S \subset V$, $1 \leq |S| \leq r-1$, such that $|N(S)| \geq f_k(n)$

5. Fix one such S and fix $T \subseteq N(S)$, $|T| = f_k(n)$;

6. $U = A(H[T], k-1)$;

7. **if** U is independent in H **return**(U);

8. **else** $H = \mathit{Reduce}(H, S)$;

9. **endwhile;**

10. **return**$(I(H, f_k(n)))$;

We claim that, given a k-colorable hypergraph H as an input, the above presented algorithm finds a large independent set. This follows from the next two propositions.

Proposition 3.3 *Algorithm $A(H,k)$ returns a subset of size $g_k(|V(H)|)$.*

Proof. By induction on k. Let $n = |V(H)|$. If $k = 1$, then the proposition is obviously true as follows from Step 2. If A returns an output U at Step 7, then U is produced at Step 6 by recursively calling A on a hypergraph on $f_k(n)$ vertices, therefore by the induction hypothesis the size of U is $g_{k-1}(f_k(n)) = g_k(n)$. Finally, if A outputs U at Step 10, then by the properties of Algorithm I, the size of U satisfies: $|U| = \frac{1}{4}(n/f_k(n))^{1/(r-1)} = g_k(n)$. $\square$

Proposition 3.4 *If H is k-colorable then $A(H,k)$ outputs an independent set in H.*

Proof. By induction on k. If $k = 1$ (i.e., H has empty edge set), then any subset of $V(H)$ is independent, therefore A returns an independent set at Step 2. Let now $k \geq 2$. Consider first the while-loop Step 4–Step 9. Note that if in the beginning of the loop execution (Step 5) H is k-colorable, then either A returns an independent set at Step 7, or by the induction hypothesis the subset T found at Step 5 spans a subhypergraph $H[T]$ which is not $(k-1)$-colorable. In the latter case, Step 8 produces an updated hypergraph $H' = \mathit{Reduce}(H,k)$, which is, by Proposition 3.1, k-colorable and such that an independent set in H' is also independent in H. Therefore, either A returns an independent set of the input hypergraph H at Step 7, or A reaches Step 10 with a hypergraph H^*, whose family of independent sets is contained in that of H. Moreover, at Step 10 for all subsets S of size $1 \leq |S| \leq r-1$ we have $|N(S)| \leq f_k(n)$. Then at this step A returns an independent set as follows from the specifications of Algorithm I. $\square$

Algorithm A is relatively effective for small values of the chromatic number k. Similarly to Wigderson's paper, when k is large we will switch to the following algorithm for finding an independent set. It is worth noting that the idea of partitioning the vertex set of a k-colorable hypergraph

H on n into bins of size $k\log_k n$ and performing an exhaustive search for an independent set of size $\log_k n$ in each bin is due to Berger and Rompel [3].

Let

$$h_k(n) = \log_k n = \log n / \log k \ .$$

Algorithm $B(H,k)$
Input: An integer $k \geq 2$ and a hypergraph $H = (V,E)$.
Output: A subset U of $V(H)$ of size $|U| = h_k(|V(H)|)$.

1. $n = |V(H)|$; $h = h_k(n)$;
2. $l = \lfloor n/hk \rfloor$;
3. Partition $V(H)$ into sets $V_1,\ldots,V_l$ where $|V_1| = \ldots = |V_{l-1}| = hk$ and $hk \leq |V_l| < 2hk$;
4. **for** $i = 1$ **to** l
5.   **for** each subset U of V_i of size $|U| = h$
6.   **if** U is independent in H **then return**(U);
7. **return** an arbitrary subset of $V(H)$ of size h;

Proposition 3.5 *For a k-colorable hypergraph H on n vertices Algorithm B outputs an independent set of size $h_k(n)$, in time polynomial in n.*

Proof. If H is k-colorable it contains an independent set I of size $|I| \geq n/k$. Then for some $1 \leq i \leq k$ we have $|I \cap V_i| \geq |I|/l \geq n/kl \geq h_k(n)$. Checking all subsets of V_i of size $h_k(n)$ will reveal an independent set of size $h_k(n)$. The number of subsets of size $h_k(n)$ to be checked by the algorithm does not exceed $l\binom{2h_k(n)k}{h_k(n)} = (O(1)k)^{h_k(n)} = n^{O(1)}$. $\square$

As we have already mentioned above, an algorithm for finding independent sets can be easily converted to an algorithm for coloring. The idea is very simple – as long as there are some uncolored vertices (we denote their union by W), call an algorithm for finding an independent set in the spanned subhypergraph $H[W]$, color its output by a fresh color and update W. As we have two different algorithms A and B for finding independent sets, we present two coloring algorithms C_1 and C_2, using A and B, respectively, as subroutines. Since the only difference between these two algorithms is in calling A or B, we present them jointly.

Algorithms $C_1(H,k)$ **and** $C_2(H,k)$
Input: An integer $k \geq 2$ and a hypergraph $H = (V,E)$ of dimension r.
Output: A coloring of H or a message "H is not k-colorable".

1. $i = 1$; $W = V$;
2. **while** $W \neq \emptyset$
3. for $C_1(H,k)$: $U = A(H[W],k)$; for $C_2(H,k)$: $U = B(H[W],k)$;
4. **if** U is not independent in H **output** ("H is not k-colorable") and **halt**;
5. color U by color i; $i = i + 1$;
6. $W = W \setminus U$;
7. **endwhile**;
8. **return** a coloring of H;

The correctness of both algorithms C_1 and C_2 follows immediately from that of A and B, respectively. As for the performance guarantee, it can be derived from the following easy proposition, proven, for example, implicitly in the paper of Halldórsson [13].

Proposition 3.6 *An iterative application of an algorithm that guarantees finding an independent set of size $f(n) = O(n^{1-\epsilon})$ in a hypergraph H on n vertices for some fixed $\epsilon > 0$, produces a coloring of H with $O(n/f(n))$ colors.*

Corollary 3.7

1. *Algorithm $C_1(H,k)$ colors a k-colorable hypergraph H of dimension r on n vertices in at most $2n/g_k(n) = 8n^{1-\frac{1}{(r-1)(k-1)+1}}$ colors;*
2. *Algorithm $C_2(H,k)$ colors a k-colorable hypergraph H on n vertices in at most $2n/h_k(n) = 2n\log k/\log n$ colors.*

Proof. Follows immediately from Propositions 3.3, 3.4, 3.5 and 3.6. $\square$

Now, given a k-colorable hypergraph H of dimension r, we can run both algorithms C_1 and C_2 and then choose the best result from their outputs. This is given by Algorithm D below.

Algorithm $D(H,k)$
Input: An integer $k \geq 2$ and a hypergraph $H = (V,E)$ of dimension r.
Output: A coloring of H or a message "H is not k-colorable".

1. Color H by Algorithm $C_1(H,k)$;
2. Color H by Algorithm $C_2(H,k)$;
3. **if** C_1 or C_2 output "H is not k-colorable", **output**("H is not k-colorable");
4. **else return** a coloring which uses fewer colors;

Until now we assumed that the chromatic number of the input hypergraph H is given in advance. Though this is not the case for general approximation algorithms, we can easily overcome this problem, for example, by trying all possible values of k from 1 to $n = |V(H)|$ and choosing a positive output of $D(H, k)$ which uses a minimal number of colors. Denote this algorithm by $E(H)$. In particular, for $k = \chi(H)$, Algorithm C_1 produces a coloring with at most $8n^{1-\frac{1}{(r-1)(k-1)+1}}$ colors, while Algorithm C_2 gives a coloring with at most $2n \log k / \log n$ colors. Hence, the approximation ratio of Algorithm E is at most

$$\min\left\{8n^{1-\frac{1}{(r-1)(k-1)+1}}/k, \frac{2n\log k}{k\log n}\right\} .$$

The first argument of the above min function is an increasing function of k, while the second one is decreasing. For $k = (1/(r-1))\log n/\log\log n$ both expressions have order $O(n(\log\log n)^2/(\log n)^2)$. Therefore we get the following result.

Theorem 3.8 *For every fixed $r \geq 3$, coloring of hypergraphs of dimension r on n vertices is $O(n(\log\log n)^2/(\log n)^2))$ approximable.*

This bound matches the bound of Wigderson [26] for graph coloring.

4 Concluding remarks

We have discussed the problem of approximate coloring of r-uniform hypergraphs. Our main goal was to advance the state of knowledge in this problem to that of the much more studied special case of graph coloring ($r = 2$). Using a simple construction, we have shown that for every $r \geq 3$ the problem of approximate coloring of r-uniform hypergraphs is at least as hard as the graph coloring problem. This implies in particular that unless $NP \subset ZPP$, it is impossible to approximate the chromatic number of r uniform hypergraphs on n vertices within a factor of $n^{1-\epsilon}$ in polynomial time. We have also presented a general approximation algorithm with approximation ratio $O(n(\log\log n)^2/(\log n)^2))$ for r-uniform hypergraphs on n vertices.

Despite some progress achieved in this paper, many problems remain open and seem to be quite interesting. One of them is to develop a general approximation algorithm, aiming to match the approximation ratio $O(n(\log\log n)^2/(\log n)^3)$ of the best known graph coloring algorithm due to Halldórsson [13]. Another interesting problem is to come up with a more involved algorithm for the case of r-uniform 2-colorable hypergraphs for $r \geq 4$. Also, new ideas for the case of 3-uniform 2-colorable hypergraphs may lead to a further improvement in this case.

Acknowledgment. We would like to thank Thomas Hofmeister for bringing paper [15] to our attention.

References

[1] N. Alon, P. Kelsen, S. Mahajan and H. Ramesh, *Coloring 2-colorable hypergraphs with a sublinear number of colors*, Nordic J. Comput. 3 (1996), 425–439.

[2] N. Alon and J. H. Spencer, **The probabilistic method**, Wiley, New York, 1992.

[3] B. Berger and J. Rompel, *A better performance guarantee for approximate graph coloring*, Algorithmica 5 (1990), 459–466.

[4] A. Blum, *New approximation algorithms for graph coloring*, J. ACM 31 (1994), 470–516.

[5] A. Blum and D. Karger, *An* $\tilde{O}(n^{3/14})$*-coloring algorithm for 3-colorable graphs*, Inform. Process. Lett. 61 (1997), 49–53.

[6] R. Boppana and M. M. Halldórsson, *Approximating maximum independent sets by excluding subgraphs*, Bit 32 (1992), 180–196.

[7] J. Brown, *The complexity of generalized graph colorings*, Discrete Applied Math. 69 (1996), 257–270.

[8] J. Brown and D. Corneil, *Graph properties and hypergraph colourings*, Discrete Math. 98 (1991), 81–93.

[9] H. Chen and A. Frieze, *Coloring bipartite hypergraphs*, in: Proc. 5^{th} Intern. IPCO Conference, Lecture Notes in Comp. Sci. 1084 (1996), 345–358.

[10] I. Dinur, O. Regev and C. Smyth, *The hardness of 3-uniform hypergraph coloring*, to appear in Proc. 43^{rd} IEEE FOCS, IEEE (2002).

[11] U. Feige and J. Kilian, *Zero knowledge and the chromatic number*, in: Proc. 11^{th} Annual IEEE Conf. on Computational Complexity, 1996.

[12] V. Guruswami, J. Håstad and M. Sudan, *Hardness of approximate hypergraph coloring*, Proc. 41^{st} IEEE FOCS, IEEE (2000), 149–158.

[13] M. M. Halldórsson, *A still better performance guarantee for approximate graph coloring*, Inform. Process. Lett. 45 (1993), 19–23.

[14] J. Håstad, *Clique is hard to approximate within* $n^{1-\epsilon}$, in: Proc. 37^{th} IEEE FOCS, IEEE (1996), 627–636.

[15] T. Hofmeister and H. Lefmann, *Approximating maximum independent sets in uniform hypergraphs*, Proc. of the 23^{rd} International Symposium on Mathematical Foundations of Computer Science, Lecture Notes in Computer Science 1450, Springer Verlag (1998), 562-570.

[16] D. S. Johnson, *Worst case behaviour of graph coloring algorithms*, in: Proc. 5^{th} S.E. Conf. on Combinatorics, Graph Theory and Computing, Congr. Numer. 10 (1974), 512–527.

[17] D. Karger, R. Motwani and M. Sudan, *Approximate graph coloring by semidefinite programming*, in: Proc. 35^{th} IEEE FOCS, IEEE (1994), 2–13.

[18] P. Kelsen, S. Mahajan and H. Ramesh, *Approximate hypergraph coloring*, in: Proc. 5^{th} Scandinavian Workshop on Algorithm Theory, Lecture Notes in Comp. Sci. 1097 (1996), 41–52.

[19] S. Khanna, N. Linial and M. Safra, *On the hardness of approximating the chromatic number*, in: Proc. 2^{nd} Israeli Symposium on Theor. Comp. Sci., IEEE (1992), 250–260.

[20] S. Khot, *Hardness results for approximate hypergraph coloring*, in: Proc. 34^{th} ACM STOC, ACM Press (2002), 351–359.

[21] S. Khot, *Hardness results for coloring 3-colorable 3-uniform hypergraphs*, to appear in Proc. 43^{rd} IEEE FOCS, IEEE (2002).

[22] M. Krivelevich, R.Nathaniel and B. Sudakov, *Approximating coloring and maximum independent set in 3-uniform hypergraphs*, J. of Algorithms 41 (2001), 99–113. A preliminary version appeared in: Proc. of the 12^{th} Annual ACM-SIAM SODA, ACM Press (2001), 327–328.

[23] M. Krivelevich and B. Sudakov, *Approximate coloring of uniform hypergraphs*, Proc. of the 6^{th} Annual European Symposium on Algorithms (ESA'98), Lecture Notes in Computer Science 1461, Springer Verlag (1998), 477-489.

[24] L. Lovász, *Coverings and colorings of hypergraphs*, in: Proc. 4^{th} S.E. Conf. on Combinatorics, Graph Theory and Computing, 1973, Utilitas Math., 3–12.

[25] K. Phelps and V. Rödl, *On the algorithmic complexity of coloring simple hypergraphs and Steiner triple systems*, Combinatorica 4 (1984), 79–88.

[26] A. Wigderson, *Improving the performance guarantee for approximate graph coloring*, J. ACM 30 (1983), 729–735.