Linear time coloring of T-free hypergraphs

Simi Haber[1]

Department of Mathematics, Bar-Ilan University, Ramat-Gan 5290002, Israel, simi@math.biu.ac.il

Abstract. The aim of this paper is to provide a structural claim for general hypergraphs, extending a theorem of Loh for uniform hypergraphs. Given a hypertree with t edges and a not necessarily uniform hypergraph H, we propose a linear time algorithm that either colors H with at most t colors or finds a homomorphic copy of T in H.

1 Introduction

Hypergraph coloring is the assignment of colors to vertices of a hypergraph such that every hyperedge contains at least two colors. The minimal number of colors needed to color a hypergraph H is called the *chromatic number* of H, denoted $\chi(H)$. Hypergraph coloring is studied both for its theoretical importance and as a framework for applications. In many applications, e.g. constraint satisfaction, declustering and analysis of many-body systems, the resulting hypergraph is not uniform.

Finding $\chi(H)$ is known to be NP-hard. In fact even determining if a hypergraph is 2-colorable is NP-complete [1][1].The difficulty of hypergraph coloring suggests looking for approximation results, however this direction also proved to be extremely hard. Currently the best polynomial time algorithms for coloring 2-colorable 3-uniform hypergraphs can only guarantee finding a coloring with $O\left(n^{1/5}\right)$ colors [2], n being the number of vertices. In fact it is known [3] that even for 3-uniform hypergraphs, for any constant $c \geq 2$ it is NP-hard to color a 2-colorable k-uniform hypergraph with c colors. The case of k-uniform hypergraphs for $k \geq 4$ was known before [4]. Non-uniform hypergraphs received less algorithmic attention, but some bounds are known [5].

Another approach may be to restrict our attention to hypergraphs satisfying some property. As in the case of graph coloring [6] one may consider hypergraphs not containing some given subhypergraphs. We will consider hypergraphs that are T-free where T is a hypertree defined[2] as a linear connected hypergraph with no cycles. In [7] the authors took this approach and were able to show (among other results) that if T is an r-uniform hypertree with t edges then every T-free hypergraph has chromatic number at most $2(r-1)(t-1)+1$. Soon after Loh improved this result with the following theorem [8]:

Theorem 1 (Loh, 2009). *Every r-uniform hypergraph with chromatic number greater than t contains every r-uniform hypertree with t edges as a subhypergraph.*

This is tight since the complete r-uniform clique on $(r-1)t$ vertices has chromatic number t, but it cannot contain a copy of an r-uniform hypertree since such a tree has $(r-1)t+1$ vertices.

[1] It is not hard to alter the proof and show that the decision problem remains NP-complete even for 3-uniform hypergraphs.

[2] Our definition follows [7, 8] but is different than [9]

Our contribution stems from a simple insight: Hypertrees are linear hypergraphs, and hence uniformity does not play a role. This leads to a generalization of Theorem 1 for non-uniform hypergraphs, yielding a rare structural result for general hypergraphs.

2 General hypergraphs

We need some notation before stating the result. Let $H = (V, E)$ be a hypergraph. If the intersection of any two edges of E has at most one vertex, we call H *linear*. A *path* in a hypergraph is a sequence $(v_0, e_1, v_1, \dots, e_\ell, v_\ell)$ where $v_{i-1}, v_i \in e_i$ for all $1 \le i \le \ell$ and all vertices and edges are distinct. A *cycle* is a path where $v_0 = v_\ell$. A hypergraph is *connected* if there is a path between any two vertices.

A *hypertree* is a connected linear hypergraph with no cycles. Let $T = [V, E]$ be a hypertree. We define the set of *joints*, $J(T)$, to be the set of vertices incident to more than one edge. The *joint reduction* of T, denoted $T|_{J(T)}$ (or sometimes simply $T|_J$), is a multi-hypertree. Its vertex set is $J(T)$, and its edge set is $\{e \cap J(T) | e \in T\}$. Notice that the linearity of T implies that multiple edges in $T|_J$ are always singletons. If $T_1|_{J(T_1)} \cong T_2|_{J(T_2)}$ as multi-hypergraphs we say that T_1 and T_2 are *joint isomorphic*. Notice that if T_1 and T_2 are joint isomorphic, they have the same number of edges. If no subhypergraph of H is joint isomorphic to T we say that H is T*-free*. In the last definition we forbid both induced and non-induced subhypergraphs. Finally, the *minimal arity* of a hypergraph H, denoted by $r(H)$ is the size of the smallest edge in H. Similarly we use $R(H)$ to denote the maximal arity of H.

We can now formulate our results.

Theorem 2. *Let H be a hypergraph with chromatic number greater than t. Then for every hypertree T with $R(T) \le r(H)$ and at most t edges, H contains a subhypertree joint isomorphic to T.*

Remark 1. If H and T are both r uniform we recover Theorem 1.

By Theorem 2, if H does not contain a certain subhypertree T with t edges as a joint isomorphic subhypergraph then H is t-colorable. In fact, in this case we can find a t-coloring in linear time:

Theorem 3. *Let H be a hypergraph with n vertices and T a hypertree with t edges such that $R(T) \le r(H)$. There is an algorithm running in time $O(nt^2)$ finding a t-coloring of H or a joint isomorphic copy of T in H.*

3 Proof

The proof closely resembles the proof of Theorem 1 from [8].

Proof of Theorem 2. Let $c \colon V(H) \to \{1, \dots, t\}$ be a (not necessarily proper) coloring of H. We may get c by, e.g., greedily coloring H and using the last color whenever no free color is available.

The algorithm has two stages. First we label edges of T. Pick an arbitrary edge of T and label it e_1. The other edges are labeled $e_2, \dots, e_t$ according to a breadth-first-search. Notice that, going along the path from e_1 to any other edge, the indices of the labels are monotonically increasing. In fact this is the only property required from the labeling. Next we color each vertex v of T by the minimal index of an edge among all edges to which v is incident. Hence e_1 is monochromatic with all vertices colored 1, and the other edges are bi-chromatic where all vertices except one are colored by the edge index, and one vertex is colored with a smaller color. We denote this coloring by c_T.

In the second stage we recolor vertices of H in order to reduce the number of monochromatic edges. As long as there are monochromatic edges we choose such an edge, denoted h_1, and map its vertices onto the vertices of e_1 (here we use the arity condition $R(T) \leq r(H)$). From this point we maintain a many-to-one mapping f from a subhypergraph H' of H to a subhypertree T' of T such that:

1. $f\colon V(H') \to V(T')$ is onto;
2. f respects the colorings. That is, $c(v_1) = c(v_2) \iff c_T(f(v_1)) = c_T(f(v_2))$;
3. f is a hypergraph homomorphism, that is, $\{v_1, \ldots, v_\ell\}$ is an hyperedge in H' if and only if $\{f(v_1), \ldots, f(v_\ell)\}$ is an hyperedge in T'.

Notice we do not require f to be injective.

H' inherits the coloring c. Colors that do not appear in H', that is, colors in $\{1, \ldots, t\} \setminus c(V[H'])$ are called *free*.

Assume $f\colon H' \to T'$ where T' is the subhypertree of T containing the edges labeled $e_1, \ldots, e_i$ for $1 \leq i < t$. For the moment, assume we have choosen f that is inclusion maximal, that is, it is not possible to extend f to a mapping from H'' (containing H' as a subhypergraph) to the subhypertree of T composed of the edges labeled $e_1, \ldots, e_{i+1}$. Notice that if $i = t$, then H' is joint isomorphic to T and the algorithm stops.

The next edge in T, e_{i+1}, intersects exactly one of $e_1, \ldots, e_i$, say e_j. Let v be the vertex in $e_j \cap e_{i+1}$ and let u be a source of v (that is, $f(u) = v$). We would have liked to recolor u. If there is a free color a such that recoloring u by a does not create a new monochromatic edge, we assign $c(u) = a$. If $i = 1$, the new coloring c has one less monochromatic edge. If there is another monochromatic edge in H, we repeat the process, otherwise we have found a proper coloring.

If $i > 1$, then after recoloring u we remove e_j and $f^{-1}(e_j)$ and update H', T' and f:

1. $E(T') = E(T') \setminus \{e_j\}$. Denote the vertices of e_j by $v_1, \ldots, v_\ell$ and assume that $v_1 \in e_k$ for some $k < j$. Then we also remove the vertices $e_j \setminus \{v_1\}$ from $V(T')$.
2. Let h_j be the source of e_j. We update $E(H') = E(H') \setminus \{h_j\}$ and $V(H') = V(H') \setminus f^{-1}(e_j \setminus \{v_1\})$.
3. Finally, we restrict f to the new H', that is, $f = f|_{H'}$.

We have removed vertices of H' along with their images, hence f is still surjective. Since u was removed from $V(H')$, the coloring c restricted to H' is unaltered, hence properties 2. and 3. above are maintained.

If we can not find a as above, i.e., if for every free color a one has that u is incident with an edge h s.t. $c(w) = a$ for all $w \in h \setminus \{u\}$, then we can extend f by mapping the vertices of $h \setminus \{u\}$ onto the vertices of $e_{i+1} \setminus \{v\}$. Such an extension is possible since $R(T) \leq r(H)$. It is immediate to verify that by construction the extended f is onto, respects the colorings and is a hypergraph homomorphism. However, this contradicts the maximality of f.

Removing the assumption of f being inclusion maximal, we get the final description of the algorithm. As long as there are monochromatic edges in H, we map such an edge to e_1. We then extend f until it is either a joint isomorphism between H' and T, or an inclusion maximal joint isomorphism between H' and T', in which case we update the coloring c.

Recoloring never creates new monochromatic edges. By the second property of f, the order over the labels of T induces an order over the colors $\{1, \ldots, t\}$. When recoloring, the new color is larger in that order. Hence the algorithm stops. This happens either when H is properly colored or when we have found a joint isomorphism between a subhypergraph of H onto T. $\square$

Looking at the innermost step in the algorithm, when a vertex is examined, it is either recolored or the mapping f is extended. Extension can be done at most $t-1$ times, and recoloring again at most $t-1$ times. All in all the second phase, and thus the algorithm, runs in $O(nt^2)$ time, which is the claim of Theorem 3.

References

1. L. Lovász, Coverings and colorings of hypergraphs, in: F. Hoffman, R. B. Levow, R. S. D. Thomas (Eds.), Proc. 4^{th} Southeastern Conf. on Comb., Vol. 8 of Congressus numerantium, Utilitas Mathematica, 1973, pp. 3–12.
2. M. Krivelevich, R. Nathaniel, B. Sudakov, Approximating coloring and maximum independent sets in 3-uniform hypergraphs, Journal of Algorithms 41 (1) (2001) 99–113.
3. I. Dinur, O. Regev, C. Smyth, The hardness of 3-uniform hypergraph coloring, Combinatorica 25 (5) (2005) 519–535.
4. V. Guruswami, J. Håstad, M. Sudan, Hardness of approximate hypergraph coloring, SIAM Journal on Computing 31 (6) (2002) 1663–1686.
5. D. A. Shabanov, Around Erdős-Lovász problem on colorings of non-uniform hypergraphs, Discrete Mathematics 338 (11) (2015) 1976–1981.
6. M. Chudnovsky, Coloring graphs with forbidden induced subgraphs, in: Proc. ICM, Vol. 4, 2014, pp. 291–302.
7. T. Bohman, A. Frieze, D. Mubayi, Coloring H-free hypergraphs, Random Structures & Algorithms 16 (1) (2010) 11–25.
8. P.-S. Loh, A note on embedding hypertrees, The Electronic Journal of Combinatorics 16, n18 (2009).
9. A. Brandstädt, F. F. Dragan, V. D. Chepoi, V. I. Voloshin, Dually chordal graphs, in: J. van Leeuwen (Ed.), Graph-Theoretic Concepts in Computer Science, Vol. 790 of Lecture Notes in Computer Science, Springer Berlin Heidelberg, Berlin, Heidelberg, 1994, pp. 237–251.