Flexible Coloring

Xiaozhou Li[a], Atri Rudra[b], Ram Swaminathan[a]

[a]*firstname.lastname@hp.com, HP Labs, 1501 Page Mill Road, Palo Alto, CA 94304*
[b]*atri@buffalo.edu, Computer Sc. & Engg. dept., SUNY Buffalo, Buffalo, NY 14260*

Abstract

Motivated by reliability considerations in data deduplication for storage systems, we introduce the problem of *flexible coloring*. Given a hypergraph H and the number of allowable colors k, a flexible coloring of H is an assignment of one or more colors to each vertex such that, for each hyperedge, it is possible to choose a color from each vertex's color list so that this hyperedge is strongly colored (i.e., each vertex has a different color). Different colors for the same vertex can be chosen for different incident hyperedges (hence the term flexible). The goal is to minimize color consumption, namely, the total number of colors assigned, counting multiplicities. Flexible coloring is NP-hard and trivially $s - \frac{(s-1)k}{n}$ approximable, where s is the size of the largest hyperedge, and n is the number of vertices. Using a recent result by Bansal and Khot, we show that if k is constant, then it is UGC-hard to approximate to within a factor of $s - \varepsilon$, for arbitrarily small constant $\varepsilon > 0$. Lastly, we present an algorithm with an $s - \frac{(s-1)k}{k'}$ approximation ratio, where k' is number of colors used by a strong coloring algorithm for H.

Keywords: graph coloring, hardness of approximation

1. Introduction

Data deduplication is a storage systems technique that aims to reduce the storing of multiple copies of the same data, thereby saving storage space. The following optimization problem arises in data deduplication. A large number of data objects (binary strings for our purposes) are to be stored on some number of

disks. Each object consists of a number of blocks. For reliability considerations, blocks belonging to the same object should be stored on distinct disks so that the failure of a disk only affects one block. In storage systems, it is common that objects have identical blocks. To save space, identical blocks need not be stored multiple times. The goal is to store the fewest number of blocks without violating the "distinct-disks" rule, omitting considerations such as disk capacities.

To understand the problem better, consider the following simple example. Suppose we need to store three objects on two disks. Each object consists of two blocks: $\{A, B\}$, $\{B, C\}$, and $\{A, C\}$, respectively. Then the most economical way to store these objects is to store four blocks, say, A, C on the first disk, and B, C on the second. This placement is legitimate because the first object consists of $A|1$ (meaning block A on disk 1) and $B|2$, the second consists of $B|2$ and $C|1$, and the third consists of $A|1$ and $C|2$. It is easy to verify that storing only three blocks A, B, and C, will violate the "distinct-disks" rule stated above.

In this paper, we formulate the above problem into an optimization problem called *flexible coloring*. Section 2 presents the problem formulation and some simple observations. Section 3 presents a hardness of approximation result. We conclude with some discussion in Section 4.

2. Problem formulation

We formulate this optimization problem as the following graph-theoretic problem which we call flexible coloring. Given a hypergraph H and the number of allowable colors k, a flexible coloring of H is an assignment of one or more colors to each vertex such that, for each hyperedge, it is possible to choose a color from each vertex's color list so that this hyperedge is strongly colored (i.e., each vertex has a different color). Different colors for the same vertex can be chosen for different incident hyperedges (hence the term flexible). The goal is to minimize color consumption, namely, the total number of colors assigned,

counting multiplicities. Clearly, for flexible coloring to be feasible, we need $k \geq s$, where s is the size of the largest hyperedge. It is easy to see that there is no need to assign a vertex more than s colors, because assigning s colors to a vertex gives the vertex enough "flexibility" to choose a color for any hyperedge to which the vertex belongs.

Clearly, in the above formulation, a vertex in H corresponds to a block, a hyperedge in H corresponds to an object, k corresponds to the number of disks, and color consumption corresponds to storage space consumption. In the example described earlier, the hypergraph consists of three vertices $\{A, B, C\}$, three hyperedges $\{A, B\}$, $\{B, C\}$, $\{C, A\}$, and $s = k = 2$. A legitimate flexible coloring is coloring A with color 1, B with color 2, and C with colors 1 and 2. The total color consumption is four.

For $s = 2$, computing the optimal flexible coloring is equivalent to the problem of finding the maximum k-colorable induced subgraph, but for general s, we are not aware of a similar problem.

Finally, we point out that given a valid assignment of colors to every vertex, a valid assignment of colors to the endpoints of an hyperedge can easily be obtained by solving a bipartite maximum matching problem (where the endpoints form the left vertices, the colors $\{1, \ldots, k\}$ form the right side, and (v, c) is an edge if the color c is assigned to the end-point v).

3. Hardness of approximation

Flexible coloring is NP-hard, because it contains graph coloring as a special case. To see this, take an instance of the graph coloring problem (G, k), where we wish to determine if graph G is k-colorable. If flexible coloring can correctly answer the question "Can G be flexibly colored with k allowable colors and consume only $|V(G)|$ colors?", then it can solve the graph coloring problem, because the requirement of consuming only $|V(G)|$ colors forces flexible coloring to assign exactly one color to each vertex.

Following the terminology for graph coloring, we say a hypergraph H is (k, ℓ)-

flex-colorable if there is a flexible coloring for H that uses k allowable colors and consumes at most ℓ colors, counting multiplicities. Given H and k, we call the smallest number ℓ such that H is (k, ℓ)-flex-colorable the *consumption number* of H and k, denoted by $\varphi(H, k)$. Observe that $\varphi(H, k) \geq n$, where $n = |V(H)|$, because every vertex is assigned at least one color. On the other hand, H is trivially $(k, sn-(s-1)k)$-flex-colorable, because we can assign one distinct color to (any) k vertices and s colors to the remaining $n-k$ vertices. These two simple observations indicate that flexible coloring is trivially $(s-\frac{(s-1)k}{n})$-approximable.

In what follows, we show that, if k is constant, then s is essentially the best ratio that can be achieved. To prove this, we use the following recent result by Bansal and Khot.

Theorem 1. *([1]) Assuming the Unique Games Conjecture (UGC), for any integer $s \geq 2$ and arbitrary constants $\alpha, \beta > 0$, given an s-uniform hypergraph $H = (V, E)$, distinguishing between the following cases is NP-hard:*

(YES): there exists disjoint subsets $V_1, \ldots, V_s \subseteq V$, satisfying $|V_i| \geq \frac{1-\alpha}{s} \cdot |V|$ and such that no hyperedge contains at least two vertices from some V_i.

(NO): every vertex cover has size at least $(1 - \beta) \cdot |V|$.

We now state and prove the main theorem of this section.

Theorem 2. *If k is a constant, then it is UGC-hard to approximate the consumption number to within a factor of $s-\varepsilon$, for arbitrarily small constant $\varepsilon > 0$.*

Proof. Recall that in order for flexible coloring to be feasible, we need $s \leq k$. Therefore, if k is given to be constant, then so is s. Consider an s-uniform hypergraph $H = (V, E)$. Let $n = |V|$. Suppose H is in the YES case in Theorem 1. Let $V' = V \setminus (V_1 \cup V_2 \cup \cdots \cup V_s)$. Since $|V_i| \geq \frac{1-\alpha}{s} \cdot n$ for all i, we have $|V'| \leq \alpha n$. Because the V_i's are disjoint and no V_i contains more than one vertex from any hyperedge, we can color the vertices in the V_i's with one color, say color i, and color those in V' with (any) s colors. Since no V_i contains more than one vertex from any hyperedge, this coloring is a legitimate flexible coloring. The total color consumption is $\sum_{i=1}^{s} |V_i| + s \cdot |V'| \leq n + (s-1) \cdot \alpha n$.

Therefore, $\varphi(H,k) \leq (1+(s-1)\alpha)n$. For any given constant ε_1, we can choose an α small enough so that $\varphi(H,k) \leq (1+\varepsilon_1)n$, because s is constant.

On the other hand, suppose H is in the NO case in Theorem 1. Recall that a vertex cover of a hypergraph is a subset of the vertices such that it contains at least one vertex in every hyperedge. Consider a flexible coloring on H. For each color set $T \subseteq \{1,2,\ldots,k\}$ such that $1 \leq |T| < s$, let V_T be the set of vertices that are colored with T. We observe that V_T does not contain any hyperedge entirely, because a hyperedge is of size s but $|T| < s$. This implies that $V \setminus V_T$ is a vertex cover. By Theorem 1, $|V_T| \leq \beta n$. Summing up over all the possible color sets T such that $1 \leq |T| < s$, we obtain an upper bound on the number of vertices that are assigned less than s colors: $\sum_{i=1}^{s-1} \binom{k}{i} \cdot \beta n$. For any constant ε_2, we can choose a β small enough so that the above summation is at most $\varepsilon_2 n$, because s and k are constants. Therefore, the number of vertices that are assigned s colors is at least $(1-\varepsilon_2)n$ (recall that no vertex needs to be assigned more than s colors), and the color consumption on these vertices (and hence on all vertices) is at least $(1-\varepsilon_2) \cdot sn$. In other words, if H is in the NO case in Theorem 1, then $\varphi(H,k) \geq (1-\varepsilon_2) \cdot sn$.

Therefore, if there is an approximation algorithm that achieves a ratio better than $\frac{1-\varepsilon_2}{1+\varepsilon_1} \cdot s$, then we will be able to tell whether H is in the YES case or the NO case, a contradiction to Theorem 1. Since ε_1 and ε_2 can be arbitrarily small, we conclude that it is UGC-hard to approximate flexible coloring to within a factor of $s-\varepsilon$, for arbitrarily small constant $\varepsilon > 0$. □

We remark that the the result above can be strengthened to obtain a hardness of approximation factor of $\frac{s}{\alpha} - \gamma$, where the optimal number of colors used is $\alpha \cdot n$ for any $1+\varepsilon' \leq \alpha \leq s-\varepsilon'$ (where $\varepsilon' = \Theta(\gamma)$). The proof follows by adding roughly $(\alpha-1)n$ "dummy" nodes to the graph produced in the reduction above and adding hyperedges so that each dummy node has the maximum possible number of incident hyperedges. (Note that in such a case the dummy nodes have to be assigned s colors.)

4. Discussion

We first remark that for $s = 2$, the optimal solution to the flexible coloring problem is the same as the maximum k-colorable induced subgraph problem. In particular, all the vertices that are assigned a single color by a flexible coloring induce a k-colorable subgraph. Using this connection one can also show that

$$n + (\chi(G) - k) \leq \varphi(G, k) \leq n + (\chi(G) - k) \cdot \frac{n}{\chi(G)},$$

where as usual, $\chi(G)$ is the chromatic number of G. Further, it can be shown that both the upper and lower bounds are tight for specific graphs.

The connection above to coloring, immediately suggests the following approximation algorithm for flex coloring a hypergraph H. Do a strong coloring on H, using any strong coloring algorithm, with no restrictions on the number of colors used. Therefore, the algorithm can use k' colors, which may be greater than k. If $k' \leq k$, we can assign one color to every vertex and finish flexible coloring with an optimal color consumption of n. If $k' > k$, then we organize the vertices into k' groups based on their colors and we sort the groups in increasing order of size. We then re-assign the vertices the first $k' - k$ groups s colors. These s colors can be any s used by group $k' - k + 1$ to group k'. It is not hard to see that this procedure produces a legitimate flexible coloring, and it consumes $\left(s - \frac{(s-1)k}{k'}\right) n$ colors, which yields an approximation ratio of $s - \frac{(s-1)k}{k'}$.

The above algorithm, which makes use of existing strong coloring algorithms, establishes a connection between the strong chromatic number and the consumption number. How well we can approximate flexible coloring now depends on how well we can approximate strong coloring. For graphs, strong coloring is just regular coloring, a well-studied problem. For example, we can use the strong coloring algorithm by Agnarsson and Halldórsson [2] or the regular coloring algorithm on graphs by Halldórsson [3]. However, since coloring is in general a hard problem to approximate, this bound is not necessarily attractive. We can also interpret the $s - \frac{(s-1)k}{k'}$ bound in terms of other graph parameters. For

example, a graph G can be greedily colored by $\Delta(G)+1$ colors, where $\Delta(G)$ is the maximum degree of G. Therefore, the above algorithm also achieves a ratio of $2-\frac{k}{\Delta(G)+1}$ for G.

As mentioned earlier, for the special case of $s=2$, flexible coloring is equivalent to the problem of maximum k-colorable induced subgraph, for which Halldórsson [4] obtained an approximation ratio of $\frac{1}{2}\left(\frac{\Delta}{k}+1\right)$ when $\Delta > k$. A simple calculation shows that if $xk' \leq \frac{1}{2}(\Delta+k)n$, where x is the size of the maximum k-colorable induced subgraph, then our bound is better, assuming that Halldórsson's algorithm does not provide a better bound for certain cases. For example, if $k' \leq \Delta+1$, then our bound is better if $x \leq \frac{n}{2}$; if $k' \leq \frac{1}{2}(\Delta+k)$, then our bound is always better because $x \leq n$.

When the hypergraph is sparse, the above algorithm can be improved. As an illustration, consider the special case where $s=2$ (i.e., graphs). We can assume that all vertices are of degree at least 1 because isolated vertices can be arbitrarily single-colored. Suppose $2m < nk$, then by a simple averaging argument, there are at least $\frac{kn-2m}{k-1}$ vertices that are of degree at most $k-1$. Observe that these low degree vertices can always be singly colored. Therefore, we can (1) exclude the low-degree vertices, (2) color the remaining induced subgraph using the above flexible coloring algorithm, and (3) add back the low-degree vertices and single color them. This algorithm results in at most $\left(1-\frac{k}{k'}\right)\cdot\frac{2m-n}{k-1}$ vertices (as opposed to the earlier $\left(1-\frac{k}{k'}\right)n$) being doubly colored. We note that step (1) above can be repeated multiple times.

We conclude by remarking that there is room for improvement for our algorithm. For example, re-assigning s colors to a vertex is brute force. A more refined method would be to first analyze whether a smaller color set is possible (e.g., a vertex with at most $k-1$ neighbors can be singly colored).

References

[1] N. Bansal, S. Khot, Inapproximability of hypergraph vertex cover and applications to scheduling problems, in: Proceedings of the 37th International

Colloquium on Automata, Languages and Programming (ICALP), 2010, pp. 250–261.

[2] G. Agnarsson, M. M. Halldórsson, Strong colorings of hypergraphs, in: Proceedings of the Third Workshop on Online and Approximation Algorithms (WAOA), 2005, pp. 253–266.

[3] M. M. Halldórsson, A still better performance guarantee for approximate graph coloring, Information Processing Letters 45 (1993) 19–23.

[4] M. M. Halldórsson, Approximating discrete collections via local improvements, in: Proceedings of the Sixth ACM-SIAM Symposium on Discrete Algorithms (SODA), 1995, pp. 160–169.