Online Coloring of Hypergraphs

Magnús M. Halldórsson[*]

February 20, 2010

Abstract

We give a tight bound on randomized online coloring of hypergraphs. The bound holds even if the algorithm knows the hypergraph in advance (but not the ordering in which it is presented). More specifically, we show that for any n and k, there is a 2-colorable k-uniform hypergraph on n vertices for which any randomized online coloring uses $\Omega(n/k)$ colors in expectation.

1 Online Hypergraph Coloring

A hypergraph $H = (V, E)$ is formed by a set V of vertices and a collection E of subsets of V. The hypergraph is *k-uniform* if each of the *hyperedges* (the elements of E) are of cardinality k. A set $S \subset V$ is an *independent set* if no edge in E is a subset of S. A *coloring* of H is a partition of V into independent sets. In the online hypergraph coloring problem, the algorithm receives in each round i, $i = 1, 2, \ldots, n$, a vertex v_i and the edges induced by (i.e., contained in) $V_i = \{v_1, v_2, \ldots, v_i\}$, and it must assign the vertex irrevocably a valid color. The objective is to minimize the number of colors.

Nagy-György and Imreh [9] gave tight bounds for deterministic online hypergraph coloring. They showed that First-Fit uses at most $\lceil n/(k-1) \rceil$ colors on k-uniform hypergraphs, while no algorithm can color every such 2-colorable hypergraph with fewer colors. It is easy to see that basically no reasonable algorithm can do much worse.

Randomized algorithms tend to attain better performance for many online problems. An oblivious adversary first chooses a graph and its ordering, and the algorithm can then use its random bits to thwart some of the worst-case nature of the instance. For ordinary graph coloring, the best performance ratio known by a randomized algorithm is $O(n/\log n)$ [5], which is considerably better than the best deterministic ratio of $O(n \log\log\log n/\log\log n)$ by a deterministic algorithm, due to Kierstead [8]. On the other hand, the best lower bound known for online graph coloring of $\Omega(n/\log^2 n)$ [7], due to Halldórsson and Szegedy, holds both for deterministic and randomized algorithms. One feature of the construction of [7] is that it holds also in a *transparent model*: immediately after the algorithm makes its assignment, the adversary reveals its intended color of the node. It also holds under other relaxations, i.e., when the algorithm is allowed logarithmic *lookahead*, buffering, or a constant fraction of recolorings.

[*] School of Computer Science, Reykjavik University, 101 Reykjavik, Iceland.

In a stronger model (see, e.g., [4]), the graph is given in advance to the algorithm, but not the order in which the vertices are presented. Namely, after all n vertices have been presented, the presented graph must be isomorphic to the graph given in advance — the difficulty lies in the fact that this isomorphism is not given. For this model, a lower bound of $\Omega(n/\log^2 n)$ also holds [6].

A different version of online coloring a known graph was considered by Bartal, Fiat, and Leonardi [3]. In their model, each presented vertex is identified on arrival with one of the underlying vertices. The difficulty then lies in the fact that the algorithm does not know which subset of the vertices will be presented. Also, the adversary is charged only for the chromatic number of the presented subgraph, as opposed to that of the underlying graph in our model. They prove a lower bound of $\Omega(n^{1-\log_4 3}) \approx \Omega(n^{0.2})$ and an upper bound of $O(\sqrt{n})$.

A related problem is the online independent set problem in hypergraphs. Alon, Arad and Azar [1] gave a tight $\theta(n/k)$ bound on the performance of deterministic and randomized algorithms.

Our results We extend the $\Omega(n/k)$ lower bound for deterministic online hypergraph coloring to randomized online algorithms against an oblivious adversary. The hypergraphs constructed are 2-colorable. The bound holds also in the transparent model, where the algorithm is immediately informed of the true color of the node, and in the known graph model, where the algorithm is given the full graph in advance.

We can also view these constructions as giving stronger bounds for online independent sets in hypergraphs, in that we restrict the given graphs to be 2-colorable.

2 Lower Bound Construction

By Yao's lemma [10], in order to prove a lower bound for randomized algorithms, it suffices to give a probability distribution for which any deterministic algorithm exceeds that bound.

Theorem 1 *For any n and $k \geq 3$, there is a 2-colorable k-uniform hypergraph $H_{n,k}$ on n vertices with the following property: There is a probability distribution over the vertex orderings of $H_{n,k}$ such that any deterministic online coloring algorithm uses at least $n/(2k)$ colors in expectation.*

This implies our main result.

Corollary 2 *Let $k \geq 3$. The performance ratio of any randomized online algorithm for coloring k-uniform hypergraphs is $\Omega(n/k)$, even if the hypergraph is known in advance.*

Construction For any $k \geq 3$ and any even number $n \geq k-2$, we consider the following hypergraph $H_{n,k}$ with vertex set $V_n = \{v_0, v_1, \ldots, v_{n-1}\}$. Let A and B denote the sets of even- and odd-numbered vertices, respectively. A *mixed q-set* is a set of q vertices from both A and B. The graph $H_{n,k}$ is constructed recursively as follows. The base case $H_{k',k}$, with $k' = 2\lceil (k-2)/2 \rceil$, has no edges and $k'/2 = \lceil (k-2)/2 \rceil$ vertices in each of A and B. Inductively, for $n > k'$, the graph $H_{n,k}$ consists of the graph $H_{n-2,k}$, the vertices v_{n-2} and

v_{n-1}, and the edges $\{v_{n-2}\} \cup S$ and $\{v_{n-1}\} \cup S$, for each mixed $(k-1)$-set S in V_{n-2}. This completes the construction. Observe that the vertices v_{n-2} and v_{n-1} are *indistinguishable* in $H_{n,k}$ (i.e., they have the same set of neighbors). Observe that A and B form a 2-coloring, since each edge is mixed.

The ordering of $H_{n,k}$ that is presented to the algorithm is in the form of two sequences from V_n. The first sequence contains from each pair v_{2q-2}, v_{2q-1} one of the vertices chosen at random, while the second sequence contains the remaining vertices. Formally, let $b_0, b_1, \ldots, b_{N-1}$ be random bits and $\overline{b_0}, \overline{b_1}, \ldots, \overline{b_{N-1}}$ be their complements, where $N = n/2$. The presentation of the graph is given by the vertex ordering

$$X = \langle v_{0+b_0}, v_{2+b_1}, \ldots, v_{2(N-1)+b_{N-1}}, v_{0+\overline{b_0}}, v_{2+\overline{b_1}}, \ldots, v_{2(N-1)+\overline{b_{N-1}}} \rangle \ .$$

In our analysis, we only consider the actions of the algorithm on the first half of the node set.

Analysis We shall refer to the color classes used by the online algorithm as *bins*, and the colors A and B of the construction as *types.* A bin is *mixed* if it contains nodes of both types.

We first describe the intuition behind the proof. Once a bin becomes mixed, it can only receive additional nodes if it contains fewer than $k-1$ nodes. Recall that each pair v_{2t-1} and v_{2t} of vertices is indistinguishable. Thus, when a node v is added to a non-empty bin, the probability that a bin becomes mixed, is at least the probability that v has different type than the first node in that bin, which is $1/2$. The number of nodes that can be added until a bin becomes mixed is a geometric random variable with expected value 2. Thus, the expected size of any bin, or the expected number of vertices that a bin receives before it becomes unusable, is at most $\max\{k-1, 3\} \leq k$. The expected number of bins is then at least N/k, where $N = n/2$ is the number of vertices in the first sequence.

Formally, let ALG be a particular deterministic online coloring algorithm. We shall show that the expected number of colors used by ALG on X, over the random choices $b_0, b_1, \ldots, b_{N-1}$, is at least $n/(2k)$. We say that a mixed bin with $t < k$ vertices has $k-1-t$ *unused slots*, since any set of $k-1-t$ vertices can be added to the bin.

We use the *potential function* $\Phi = n_B + (k-1) \cdot n_M - s_M$, where n_B is the number of non-empty bins, n_M is the number of mixed bins, and s_M is the total number of unused slots. Recall that a mixed bin can only receive additional nodes if it contains fewer than $k-1$ nodes; hence, each mixed bin contains at most $k-3$ unused slots. Note that $n_B \leq \Phi \leq n_B + (k-1)n_M \leq k \cdot n_B$, and initially $\Phi = 0$. We shall show that Φ increases by at least 1, in expectation, in the first N rounds. This yields the theorem.

Consider the step when a node v is presented and is assigned by the algorithm to a bin. Let $\Delta\Phi$ be the change in potential by the coloring of v. There are three possible cases for the algorithm's assignment of the vertex v:

1. v is assigned to a new bin.
   Then, $\Delta\Phi = 1$, since n_B increases by 1, while n_M and s_M stay unchanged.

2. v is assigned to a mixed bin.
   Then, $\Delta\Phi = 1$, since s_M decreases by 1, while n_B and n_M stay unchanged.

3. v is assigned to an unmixed bin.
   Then, with probability $1/2$, v is of the same type (A or B) as the other nodes in the bin, in which case $\Delta\Phi$ remains unchanged. Also, with probability $1/2$, v is of different type than other nodes in the bin. In this case, n_M goes up by one, s_M goes up by at most $k-3$, and thus $(k-1)n_M - s_M$ goes up by at least two. Hence, the expected value of $\Delta\Phi$ is at least 1.

It follows that after the first N vertices have been presented, the expected potential Φ is at least N. Since $\Phi \leq kn_B$, it follows that the expected number n_B of bins used by the algorithm is at least $N/k = n/(2k)$.

Remark: In the case of unknown hypergraphs (the usual online model), the factor 2 can also be omitted from the above argument by only giving the first half of the set X and then stopping the input.

Corollary 3 *Any randomized online algorithm uses at least n/k colors in expectation on 2-colorable k-uniform hypergraphs, when the graph is not known in advance.*

References

[1] N. Alon, U. Arad, and Y. Azar. Independent sets in hypergraphs with applications to routing via fixed paths. In *Proc. 2nd Intl. Workshop on Approx. Alg. for Combinatorial Optimization Problems (APPROX)*, pp. 16–27, 1999.

[2] A. Bar-Noy, P. Cheilaris, S. Olonetsky, and S. Smorodinsky. Online conflict-free coloring for geometric hypergraphs. *Combinatorics, Probability and Computing*, Published online, 11 Dec 2009.

[3] Y. Bartal, A. Fiat, and S. Leonardi. Lower bounds for on-line graph problems with application to on-line circuit and optical routing. In *Proc. 28th Ann. ACM Symp. on Theory of Computing (STOC)*, pages 531–540, 1996.

[4] A. Gyárfás and J. Lehel. On-line and first-fit colorings of graphs. *J. Graph Theory* 12(2): 217–217, 1988.

[5] M. M. Halldórsson. Parallel and on-line graph coloring. *J. Algorithms*, 23:265–280, 1997.

[6] M. M. Halldórsson. Online coloring known graphs. *Electronic J. Combin.*, R7, 2000.

[7] M. M. Halldórsson and M. Szegedy. Lower bounds for on-line graph coloring. *Theoretical Computer Science*, 130:163–174, August 1994.

[8] H. Kierstead. On-line coloring k-colorable graphs. *Israel J. Math.*, pages 93–104, 1998.

[9] J. Nagy-György and Cs. Imreh. Online hypergraph coloring. *Inf. Proc. Let.*, 109(1):23–26, 2008.

[10] A. C.-C. Yao. Probabilistic computations: Toward a unified measure of complexity. In *Proceedings of the 18th Annual ACM Symposium on Theory of Computing (STOC)*, pages 222–227, 1977.