Online Algorithms for Edge Coloring

Tomás Feder*Rajeev Motwani†Rina Panigrahy‡

Abstract

We study the problem of online edge coloring that arises as an important application in switch fabric scheduling on routing and switching devices. We give four simple randomized algorithms for edge-coloring graphs of maximum degree Δ, for which the optimum number of colors achievable is known to be $\Delta + 1$. The first algorithm randomly selects non-adjacent edges at random until a maximal matching is obtained, colors this matching 1 and removes the matching, then repeats for colors $2, 3, \ldots$. The second algorithm randomly selects edges and colors each edge with the least available color, for colors $1, 2, 3, \ldots$. Both algorithms are shown to require at most 1.9615Δ colors with high probability, smaller by a constant factor from the obvious $2\Delta - 1$, assuming $\Delta > c \log n$ for a large constant $c > 0$. Both algorithms will use at least 1.2320Δ colors with high probability, even for trees with all vertices at distance at most two from the root. The third algorithm chooses a palette of colors $1, 2, 3, \ldots$ and colors the edges in an arbitrary order with a valid color chosen at random from valid colors from the palette. This algorithm is not known to require more than $\Delta + O(\sqrt{\Delta \log n})$ colors, is shown to meet this bound for trees, and if each edge belongs to a cut having at most r edges meets the bound $\Delta + r + O(\sqrt{\Delta \log n})$, where $r = 1$ is the case of trees. The fourth algorithm adds flexibility to the third algorithm by choosing the order of edges at random as in the first two algorithms, so may meet the conjectured $\Delta + O(\sqrt{\Delta \log n})$ bound. We show for the fourth algorithm a bound 1.991Δ if $\Delta > c \log n$ for a large constant $c > 0$. If for some $\epsilon > 0$, every edge of G belongs to a cut whose degree is at most $\Delta^{1-\epsilon}$, then there exist $c, \delta > 0$ such that if $\Delta > (\log n)^c$ then the fourth algorithm may use a palette with $\Delta(1 + \Delta^{-\delta})$ colors.

1 Introduction

It is known that if a graph G with n vertices and m edges has maximum degree Δ, then it can be edge colored with $\Delta + 1$ colors, or in the bipartite case with Δ colors.

In the problem of online edge coloring, we are given a graph G and edges are coming in online. We are supposed to color the edges as they come, with the minimum number of total colors used. This problem arises as an important model in the design of switch fabric scheduling algorithms. A switch fabric can be modeled as a bipartite graph, where the input and output ports can be thought of as nodes, and packets can be thought of as edges connecting the source input port to the destination output port. All packets of one color as scheduled in one time step. Minimizing the total number of time steps for switching packets naturally corresponds to the online edge coloring problem [1, 2, 3, 4, 5]

We give here the first algorithm for the general algorithm that improves by a constant factor over the obvious $2\Delta - 1$ bound on graphs with maximum degree Δ. The algorithm that achieves

*268 Waverley Street, Palo Alto, CA 94301. Email: `tomas@theory.stanford.edu`

†Department of Computer Science, Stanford University, Stanford, CA 94305. Supported in part by NSF Grants IIS-0118173, EIA-0137761, and ITR-0331640, and grants from Microsoft, SNRC, and Veritas. Email: `rajeev@cs.stanford.edu`

‡Department of Computer Science, Stanford University, Stanford, CA 94305. Email: `rinap@cs.stanford.edu`

$2\Delta - 1$ colors is given by coloring each edge that comes with the least numbered available color. The bound follows from the fact that $\Delta - 1$ incompatible colors may occur at either side of each edge before the edge is colored, so that $(\Delta - 1) + (\Delta - 1) + 1 = 2\Delta - 1$.

2 Edge coloring with $1.2320\Delta < C < 1.9615\Delta$ colors

We introduce two simple new algorithms. Both algorithms are greedy, that is, color edges in some arbitrary given order. The potential colors are ordered $1, 2, \ldots, 2\Delta - 1$, and when an edge is colored, the lowest available color for that edge (different from the at most $2(\Delta - 1)$ colors of adjecent edges) is used.

First algorithm: The edges are ordered at random, and we greedily assign color 1 until a maximal matching is colored 1, then we start over with a random order of the remaining uncolored edges and we greedily assign color 2 until a maximal matching of the remaining edges is colored 2, and so on with the remaining edges and maximal matchings for colors $3, 4, \ldots$ in order until all edges are colored with at most $2\Delta - 1$ colors.

Second algorithm: The edges are ordered at random only once, and each edge in turn is colored with the least valid color out of $1, 2, \ldots, 2\Delta - 1$.

For both algorithms, we show that there exists a constant c such that if $\Delta > c \log n$ then with high probability the number of colors used is at most 1.9615Δ, fewer than $2\Delta - 1$ by a constant factor.

Also for both algorithms, we show that the expected number of colors used in some cases of trees is at least 1.2320Δ.

Theorem 2.1 *The first algorithm uses at most* 1.9615Δ *colors.*

Proof. Let $e = (x, y)$ be an edge to be assigned the last color. Let $x_1, \ldots, x_a$ be the neighbors of x other than y, and let $y_1, \ldots, y_b$ be the neighbors of y other than x.

Assume first the graph is triangle-free. Let $r = 0.2975$, and consider coloring one of the edges incident to one of the x_i with color $j \leq r\Delta - (\Delta - a)$. This happens when an edge incident to one of the x_i is chosen. We shall assume that to choose such an edge, we choose a pair of edges $f = (x, x_i)$, $g = (x_i, x')$ with $x' \neq x$ and color f with probability $1/(\Delta + 1)$ or color g with probability $1 - 1/(\Delta + 1)$. As there are at most $\Delta - 1$ choices of g, the probability that f is colored with these pairs is at most $(\Delta - 1)/(\Delta + 1)$, as opposed to the larger probability $\Delta/(\Delta + 1)$ for the edges g. For the remaining probability $1/(\Delta + 1)$ the edge f is chosen by itself to be colored. Let $s = 0.0385$ and $t = (1/2)(r + s) = 0.1680$, and suppose fewer than $t - (\Delta - a)$ edges $f = (x, x_i)$ had previously been colored. Then there are potentially $a - (1/2)(r + s) + (\Delta - a) = \Delta - (1/2)(r + s)$ edges that can be colored j. The probability that none will be colored is given by the recurrence $h(z) = (1 - 1/(\Delta + 1))h(z - 1)$, $h(0) = 1$ for $z = \Delta - (1/2)(r + s) = (1 - t)\Delta$ for $t = (1/2)(r + s) = 0.1680$, giving $h(z) = (1 - 1/(d + 1))^z = e^{t-1}(1 + o(1))$. Thus out of the $r\Delta$ colors, we can expect to use $t'\Delta$ colors plus the missing $\Delta - a$ edges, for $t' = r(1 - e^{t-1}) > t$. For any $c' > 0$, if $\Delta > c \log n$ for c large enough, the probability that fewer than t colors plus missing edges will happen is at most $e^{-c''\delta} < 1/n^{c'}$ by the Chernoff bound. A similar anaysis holds for y, b, y_i as for x, a, x_i, so out of the first $r\Delta$ colors we repeat $(2t - r)\Delta = s\Delta$ colors at x and y, so by the time $e = (x, y)$ is colored we have only used $(2 - s)\Delta = 1.9615\Delta$ colors completing the proof when there are new triangles.

If there are triangles in G, then $g = (x_i, x')$ may have $x' = x_{i'}$ for some $i' \neq i$, forming a triangle $xx_ix_{i'}$. In that case we may choose g either from the side of x_i or the side of $x_{i'}$, so we choose f with probability $2/(\Delta + 1)$ and g with probability $1 - 2/(\Delta + 1)$. So if out of the at most $\Delta - 1$

edges (x_i, x'), we have p with $x' = x_j$ for some x_j and q other edges, $p + q = \Delta - 1$, chosen as from before, then choosing p' so that $p'2(1 - 2/(\Delta + 1)) = 1 - 1/(\Delta + 1)$ we have for f probability $p/(\Delta + 1) + qp'2/(\Delta + 1) = p/(\Delta + 1) + q(1 - 1/(\Delta + 1))/(\Delta - 1) \leq 1 - 1/(\Delta + 1)$ as before, completing some probability for (x, x_i) chosen alone. The recurrence for the case where $x' = x_j$ gives $h(z) = (1 - 2/(\Delta + 1))h(z - 2) \leq (1 - 1/(\Delta + 1))^2 h(z - 2)$, solving the recurrence as from before and completing the proof for the first algorithm. □

Theorem 2.2 *The second algorithm uses at most* 1.9615Δ *colors.*

Proof. For the second algorithm, we assume again that out of the first $r\Delta$ colors for edges (x, x_i) plus $\Delta - a$ missing edges we only use $t\Delta$ colors plus missing edges, so each of the remaining $(1 - t)\Delta$ vertices have at least one (x_i, x') colored color j as the choice for a pair $(x, x_i), (x_i, x')$ for the remaining colors, and the bounds hold as in the analysis of the first algorithm. □

Lower bound:
We now show the lower bound.

Theorem 2.3 *Both the first and second algorithm have a lower bound of* 1.2320Δ *colors on tree instances.*

Proof. The instance is a tree whose root has degree Δ, and each of the children of the root has $\Delta - 1$ children that are leafs. For the first algorithm, suppose we have gone through the first k colors, and ℓ of them were not assigned to edges incident to the root. Then the probability that color $k+1$ will not be assigned to a child of the root is $(1 - 1/(\Delta - k))^{\Delta - k + \ell} = e^{-(1 + (\ell/(\Delta - k)))}(1 + o(1))$. For the second algorithm, if we consider sequentially when colors $1, 2, \ldots$ are assigned, the same bound holds. If we let $\ell(k)$ be the expected number of unassigned colors out of the first k colors, then using Chernoff's bound to show that the obtained values are with high probability close to the expected value we get $\ell(0) = 0$, $\ell(k + 1) - \ell(k) = (1 - 1/(\Delta - k))^{\Delta - k + \ell(k)}$. We estimate with Δ up to 10^9 that $\ell(\Delta) > 0.232063553\Delta$, giving the lower bound 1.232063553Δ on the expected number of colors used by both algorithms. □

3 Edge coloring with 1.991Δ colors

For the algorithms of this section, although the upper bounds are worse than those of the previous section, the algorithms could possibly require only $\Delta + O(\sqrt{\Delta \log n})$ colors.

Third algorithm: The algorithm chooses a palette of 1.991Δ colors, assumes $\Delta > c \log n$ for some sufficiently large constant $c > 0$, and given an arbitrary ordering of the edges of the graph, colors them with colors from the palette in that order by choosing a color at random at each step from the palette out of the valid colors. The algorithm succeeds with high probability.

Fourth Algorithm: The fourth algorithm is the same as the third algorithm but the order of the edges is chosen at random.

For trees, we show that if $\Delta > c \log n$ for sufficiently large c, actually $\Delta + O(\sqrt{\Delta \log n})$ colors in the palette suffice to succeed with high probability, even if the ordering of edges is chosen by an adversary.

More generally, if every edge e participates in a cut with at most r edges, and $\Delta > c \log n$ for sufficiently large c, actually $\Delta + r + O(\sqrt{\Delta \log n})$ colors in the palette suffice to succeed with high probability, even if the ordering of edges is chosen by an adversary.

Theorem 3.1 *The fourth algorithm works successfully with high probability if the palette has 1.991Δ colors.*

Proof. Choose a palette with $(1+\epsilon)\Delta$ colors. Consider an edge e and the edges E and F at both sides of e. Consider the first $\delta > 0$ fraction of the edges to be colored. By Chernoff's bound, when $\Delta > c \log n$ for sufficiently large c, this will choose a fraction between $\delta'\Delta$ and $\delta''\Delta$ edges at each vertex for some $\delta' < \delta < \delta''$ and $\delta'' - \delta'$ arbitrarily small probability. If f is one edge in E or F to be chosen, say E, with E' and F' at both sides of f, then the palettes available for edges in E' and F' are of size at least $(1+\epsilon-2\delta'')\Delta$, and these have at least $(1+\epsilon-\delta')$ colors not in F, so with probability at least $(1+\epsilon-2\delta'')/(1+\epsilon-\delta')$ such colors will not be in F, so the fraction of colors in F chosen by E' and F' is with high probability at most $2\delta''\Delta(1-(1+\epsilon-2\delta'')/(1+\epsilon-\delta'))$. Thus if f is chosen after at least $(\delta/2)\Delta$ edges have been chosen for E and for F, each of the remaining $\delta\Delta$ edges f to be chosen in E or in F, say in E, will choose a color in F with probability at least $\delta/2 - 2\delta''(1-(1+\epsilon-2\delta'')/(1+\epsilon-\delta'))$. Multiplying this with $\delta\Delta$ edges to be chosen gives at least $\mu\Delta = \delta\Delta(\delta/2 - 2\delta''^2/(1+\epsilon)$ shared colors, so only $(1+\epsilon)\Delta = (2-\mu)\Delta$ colors are needed. Setting $\mu < \delta(\delta/2 - 2\delta^2)/(2-\mu)$ suffices, giving in particular $\delta = 1/6$, and $\mu = 0.009$ suffices to give a 1.991Δ bound. □

Theorem 3.2 *In the case of trees, for the third algorithm, a palette of $\Delta + O(\sqrt{\Delta \log n})$ colors suffices.*

Proof. In the case of graphs that are trees, the colors chosen for E and F are independent immediately in C, so if the palette has $\Delta + x$ colors we have that the probability that all x extra colors in F will be chosen in E is $(\Delta/(\Delta+x))^x \approx e^{-x^2/\Delta}$, and this probability is small if $x^2/\Delta = c \log n$, that is $x = O(\sqrt{\Delta \log n})$, completing the proof. □

Theorem 3.3 *In the case where each edge e participates in a cut with r edges, a palette of $\Delta + r + O(\sqrt{\Delta \log n})$ colors suffices for the third algorithm.*

Proof. When every edge e participates in a cut with r edges, the result is the same as with trees if the edges in the cut with e are not chosen before e is chosen, as both sides are then independent. As the dependency between the two sides is only in the at most r colors used by these r edges, the other colors being chosen independently, the additional r of these r colors used in E and F is the only dependent part. The other colors may be chosen independently at random for E and F, so the bound for trees plus r is the bound obtained. □

Theorem 3.4 *If for some $\epsilon > 0$, every edge e of G belongs to a cut whose degree is at most $\Delta^{1-\epsilon}$, then there exist $c, \delta > 0$ such that if $\Delta > (\log n)^c$ then the fourth algorithm may use a palette with $\Delta(1+\Delta^{-\delta})$ colors.*

Proof. If we never choose the edges of the cut of degree $\Delta^{1-\epsilon}$, then the edges in both sides of e are chosen independently at random. As we let these edges of the cut influence the choice of colors for other edges, as a randomly chosen color may coincide with them and cause a change, this produces an edge with a bad color with respect to the color chosen when the cut edges are not selected. The proof defines $f(x)$ as the number of bad edges already chosen incident to a given edge given that x such edges have been chosen. Initially $f(0) = \Delta^{1-\epsilon}$. Furthermore $f'(x) = 2f(x)/(\Delta - x)$.

This solves to $f(x) = \left(\Delta^{1+(1-\epsilon)/2}/(\Delta - x)\right)^2$ for $x < \Delta - \Delta^{1-\delta}$, and if we consider intervals of size Δ^{δ} chosen edges we will deviate from this expected value only by a constant factor with high probability if $\Delta > (\log n)^c$. Thus the expected number of bad edges incident to any edge is at most $\Delta^{1-\delta}$, and if we choose a palette of size $\Delta + \Delta^{1-\delta}$, two random sets of size Δ on both sides of e are expected to leave uncovered $\Delta^{1-2\delta}$ colors, which account for the $\Delta^{1-3\delta}$ bad edges. $\square$

References

[1] Gagan Aggarwal, Rajeev Motwani, Devavrat Shah, and An Zhu, Switch Scheduling via Randomized Edge Coloring. IEEE Symposium on Foundations of Computer Science (FOCS03), Boston, October 2003.

[2] Cisco 12000 Gigabit Switch Router. Product overview. www.cisco.com, Feb. 2000.

[3] P. Giaccone, B. Prabhakar and D. Shah, Towards Simple, High-Performance Schedulers for High-aggregate bandwidth. Proceedings of INFOCOM 2002.

[4] N. McKeown, V. Anantharan and J. Walrand, Achieving 100 percent Throughput in an Input-Queued Switch. Proceedings of INFOCOM 1996.

[5] L. Tassiulas and A. Ephremides, Stability Properties of Constrained Queueing Systems and Scheduling for Maximum Throughput in Multiple Radio Networks, IEEE Transactions on Automatic Control, Vol. 37, No. 12, pp. 1936–1949, December 1992.