A Lower Bound for On-Line Edge Coloring

Amotz Bar-Noy [*] *Rajeev Motwani* *Joseph Naor* [†]

Computer Science Department
Stanford University
Stanford, CA 94305.

1 Introduction

Let $G = (V, E)$ be a simple graph. The chromatic number of a graph, $\chi(G)$, is the minimum number of colors needed to color the vertices such that adjacent vertices receive different colors. The chromatic index, $\chi'(G)$, is the minimum number of colors needed to color the edges such that adjacent edges receive different colors. Determining both $\chi(G)$ and $\chi'(G)$ was shown to be NP-complete [GJ]. Very little is known on polynomial time approximations to $\chi(G)$. The best bounds were achieved by [Wi] for general graphs and by [Bl] for 3-colorable graphs. However, for edge coloring the situation is better. Vizing's theorem states that $\Delta \leq \chi'(G) \leq \Delta + 1$ where Δ is the maximum degree of the graph. The proof of the theorem is constructive and a $\Delta + 1$ coloring can be obtained in polynomial time.

An algorithm is called off-line if all the input must be accessed before the output is produced. Most algorithmic research in graph theory has concentrated on off-line algorithms. In particular, the above mentioned vertex coloring algorithms and the $\Delta + 1$ edge coloring algorithm are off-line.

In contrast, an on-line algorithm has to make (partial) decisions after having seen only a subset of the input. For example, for coloring problems, there are two natural on-line models: the input is given either vertex-by-vertex, or edge-by-edge. The algorithm assigns a color to the current vertex or edge based only on past history; color assignments cannot be changed. This process can be viewed as a game between two players: an algorithm and an adversary. The adversary decides on the graph and also on the order in which the input

[*]Supported in part by a Weizmann fellowship, by contract ONR N00014-88-K-0166 and by a grant of Stanford's Center for Integrated Systems.

[†]Supported by contract ONR N00014-88-K-0166 and by a grant of Stanford's Center for Integrated Systems.

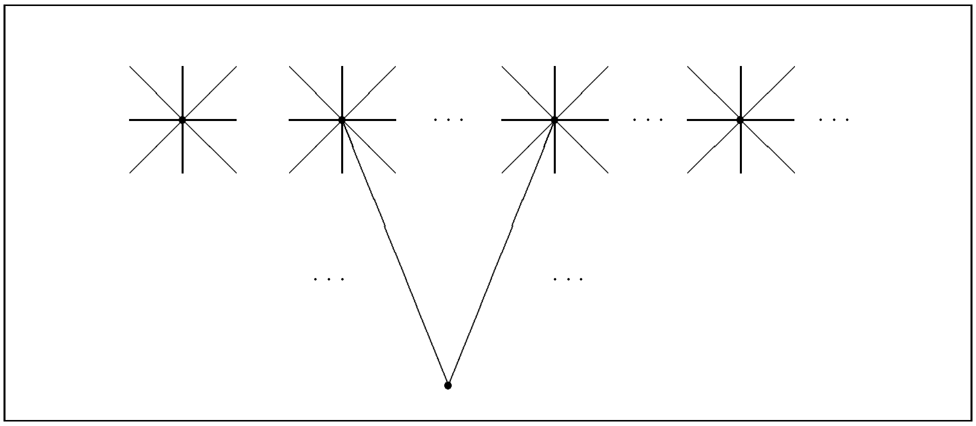

Figure 1: In the deterministic case, out of β stars there must be Δ stars with the same labeling. In the probabilistic case, with a positive probability there are Δ stars with the same labeling.

is presented to the algorithm.

The performance ratio of an on-line algorithm $\mathcal{A}$ on G is the ratio of the number of colors used by $\mathcal{A}$ to the number of colors in an optimal off-line coloring. On-line algorithms can be either deterministic or randomized. Let c be a constant. The performance ratio of a randomized algorithm is guaranteed with probability at least $\frac{1}{c}$. In the randomized case, the adversary has to commit itself to the graph before the computation starts, as opposed to the deterministic case. Otherwise, there is no point in introducing randomization.

Recently, there has been interest in on-line vertex coloring algorithms. Lovász, Saks and Trotter [LST] gave an on-line algorithm that colors any graph with at most $o(n)$ colors. Kierstead [Ki] and Gyárfás & Lehel [GL] studied on-line algorithms in various families of graphs.

The greedy coloring algorithm scans the vertices or edges one-by-one, and assigns each vertex or edges the minimum possible color. Obviously, it is an on-line algorithm. For vertex coloring, it uses at most $\Delta+1$ colors; for edge coloring it uses at most $2\Delta-1$ colors.

In this note we are concerned with on-line edge coloring. We show that the greedy algorithm, which has performance ratio 2, is optimal in the deterministic case and also in the randomized case. In both cases, we define a graph such that any on-line algorithm must use $2\Delta-1$ colors, independent of whether the graph is given edge-by-edge or vertex-by-vertex. In the deterministic case $\Delta = O(\log n)$ and in the randomized case $\Delta = O(\log^{\frac{1}{2}} n)$.

2 The lower bounds

We first describe the lower bound for the deterministic case. The idea is a pigeon-hole argument. Assume to the contrary that the algorithm can color any n-vertices graph that has maximum degree Δ with only $2\Delta - 2$ colors.

The adversary gives the algorithms β stars of degree $\Delta - 1$. In the edeg-by-edge case the order of the edges does not matter. In the vertex-by-vertex model the adversary reveals only the surraundings vertices of the stars. Each star must be colored with $\Delta - 1$ colors and we call this coloring the **labeling** of the star. If β is large enough then there must be Δ stars with the same labeling. The adversary now connect a special vertex called the **pivot** with the center of these Δ stars. Now, the algorithm has to provide Δ colors which are different from the $\Delta - 1$ colors used in the labeling. All together the algorithm needs $2\Delta - 1$ colors; a contradiction.

Let us now analyze the value of n. Define $\alpha = \binom{2\Delta-2}{\Delta-1}$, α is the number of possible distinct labeling. Then for the pigeon-hole argument $\beta = (\Delta - 1)\alpha + 1$ is sufficient. The overall number of vertices in the graph is $n = \Delta\beta + 1$. An easy calculation shows that $n < \Delta^2 4^\Delta$ and therefore in our proof Δ is of order $\log n$.

We now proceed with the probabilistic case. The difficulty arises as the adversary must commit beforehand to which stars the pivot is connected. We overcome this difficulty with a probabilistic argument for the adversary. It chooses in random Δ stars out of the β stars and connect their center with the pivot. As the algorithm cannot foresee the outcome of this random process with a positive probability, ϵ, the adversary chose Δ stars with the same labeling and then the contradiction is achieved as in the deterministic case. If we repeat this process γ times for a sufficiently large γ we can get the contradiction with high probability.

Let us now analyze the values of ϵ, γ and n. Clearly $\epsilon \geq \frac{1}{\binom{\beta}{\Delta}}$ and therefore the adversary fails with probability less than $\left(1 - \frac{1}{\beta^\Delta}\right)^\gamma$. Taking $\gamma = \Delta\beta^\Delta$ implies failure with probability at most $\frac{1}{e^\Delta}$. Now, $n = \gamma(\beta\Delta + 1)$ or $n < 4^{O(\Delta^2)}$. Hence in this counterexample Δ is of order $\log^{\frac{1}{2}} n$.

References

[Bl] A. Blum, An $\tilde{O}(n^{0.4})$-approximation algorithm for 3-coloring, Proceedings of the 21st Annual ACM Symposium on Theory of Computing, pp. 535-542, (1989).

[GJ] M. R. Garey and D. S. Johnson, Computers and intractability - a guide to the theory of NP-completeness, W. H. Freeman, San Francisco (1979).

[GL] A. Gyárfás and J. Lehel, On-line and first-fit colorings of graphs, J. of Graph Theory, **12**, pp. 217-227 (1988).

[Kr] H. A. Kierstead, The linearity of first-fit coloring of interval graphs, Siam J. Disc. Math, **1**, pp. 526-? (1988).

[LST] L. Lovász, M. Saks and W. T. Trotter, An on-line graph coloring algorithm with sublinear performance ratio, Discrete Math, **75**, pp. 319-325 (1989).

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