A Local Parallel Graph Coloring Algorithm

Michael L. Best

Thinking Machines Corporation
245 First Street
Cambridge, MA 02142
mikeb@think.com

EXTENDED ABSTRACT

Abstract

A parallel graph coloring algorithm is proposed. A local approach is employed; the graph is initially completely colored, perhaps illegally, and at each iteration some nodes recolor attempting to converge on a legal coloring. A parallel implementation of the algorithm on a Connection Machine CM-5 will be compared in speed and optimallity against other algorithms found in the literature. With a CM-5 we will be able to color graphs much larger then have been traditionally colored in the literature.

1 Overview

We will explore the coloring of large random graphs. A random graph, $\underset{\sim}{G}_{n,p}$, is a graph of order n where the adjacency of any pair of nodes is chosen independently with probability p. Johri and Matula have identified a number of motivations for the coloring of large random graphs[4]. These include (1) the use of random graph coloring to study independent random events of fixed probability in scheduling and partitioning problems, (2) the aptness of random graphs in modeling the average case behavior of a coloring algorithm, and (3) the ease of portability of test cases generated by random graphs. An edge probability of $\frac{1}{2}$ generates a set of graphs with characteristics that are the asymptotic properties of the averaging of all graphs. Thus, the properties of this class of graphs are said to hold for "almost all graphs"[1]. We intend to study large random graphs with an edge probability of $\frac{1}{2}$. Leighton has discovered a method for evolving graphs with a known chromatic number [5]. With such information we will be able to gauge the optimallity of our solutions.

A new data parallel graph coloring algorithm is proposed. The algorithm does not employ partial enumeration; at any iteration of the algorithm the entire graph is colored but perhaps invalidly. Consider coloring a random graph of order n, $\underset{\sim}{G}_{n,p}$. An initial loose upper bound, α, on the chromatic number, $\chi(G)$, is computed as the maximum degree of the graph plus one [2]. Let $\vec{x}$ be the length n ordered vector of nodes, $X_0, ..., X_{n-1}$. We choose X_0 to be a node of highest degree in $\underset{\sim}{G}_{n,p}$. X_i is that node which is most highly connected (shares the most edges) with the subgraph defined by the nodes $X_0, ..., X_{i-1}$. Ties are resolved by taking the node of highest degree (and ties within that are resolved arbitrarily) [3].

Let γ be the current number of colors used in our coloring, $c_0 ..., c_{\gamma-1}$. Initially γ is 1. Note that necessarily $\gamma \leq \alpha$. With this single color we color the entire graph. On each iteration of the algorithm every node checks each of its neighbors for a collision, that is, for an instance where it is colored identically to one of its neighbors. Consider two colliding nodes i and j. Node X_i will be chosen for recoloring iff $i > j$.Thus least connected nodes (nodes with the smallest number of edges shared with nodes of larger degree) recolor fastest.

We need to select a new color for X_i. Of the γ available colors we will compose a subset of candidate colors for the recoloring. A color will be considered a candidate if (1) it is not being used by any of X_i's neighbors, or (2) it is the color of one of X_i's neighbors and that neighbor has already determined that it will recolor. Note that because recoloring of neighbors is all taking place in parallel, we are not guaranteed that any of these candidates will be legal after the completion of this iteration. Our initial approach will be to select from the candidate subset at random. We suspect (and have some empirical evidence) that random selection will be as good (or better) then most other techniques and will protect against possible cycles in the recolorings. If the candidate subset is empty then we will select from all γ available colors at random. We are similarly motivated towards random selection here as before, however we intend to explore other selection techniques.

When the candidate subset is empty we will, at some point, wish to add additional colors to our palate as long as $\gamma < \alpha$. We will explore probabilistic points at which we increment the number of available colors. Consider a random, equiprobable, independent distribution of γ colors (in other words, a completely random coloring of the graph with γ colors). Let *S(t)* be the number of recoloring nodes at iteration ***t***. The average number of recolors is given by:

$$\utilde{S}(t) = n\left[1 - \left[1 - \frac{1}{\gamma}\right]^{np}\right] \qquad \textbf{(1)}$$

(Remember that γ is a function of *t*.) Because the number of nodes, *n*, is large we would expect the observed frequency to be close to the frequency predicted by our equation. However, we are not recoloring at random but instead are injecting “energy” into the system such as to (hopefully) reduce the number of recolors with each iteration. We propose to model this as:

$$S(t) = \utilde{S}(t) + \zeta(t) + e(t) \qquad \textbf{(2)}$$

Here $\zeta(t)$ is our model of the algorithm’s success in reducing the number of recolors with time and $e(t)$ is the error term. We will use a first-order approximation of the unknown function $\zeta(t)$ and empirically investigate its correctness.

We need to fit *S(t)* onto a line for some interval of *t*. While that line has negative slope we will not add a new color. With zero or positive slope we will increment the number of available colors, γ. Thus, with an appropriately selected interval, we will choose a moment to inject a new color when we have statistically stopped reducing recolors (and thus are no longer making progress).

We are guaranteed to coverage on some solution, though it may not be an optimal one.

References

[1] Bell Bollobas, *Random Graphs*, Academic Press, 1985.

[2] R. L. Brooks, “On Coloring the Nodes of a Network”, *Math. Proc. Cambridge Philos. Soc.*, Vol. 37, 1941.

[3] J. Randall Brown, “Chromatic Scheduling and the Chromatic Number Problem”, *Management Science*, Vol. 19, No. 4, December, Part I, 1972.

[4] Abhai Johri, David W. Matula, “Probabilistic Bounds and Heuristic Algorithms for Coloring Large Random Graphs”, Technical Report 82-CSE-6, Department of Computer Science and Engineering, Southern Methodist University, June 1982

[5] Frank Thomson Leighton, “A Graph Coloring Algorithm for Large Scheduling Problems”, *Journal of Research of the National Bureau of Standards*, Vol 84, No. 6, November-December 1979.