A Backtracking Correction Heuristic for Improving Performance of Graph Coloring Algorithms *

S. Bhowmick and P. Hovland

Mathematics and Computer Science Division, Argonne National Laboratory,
9700 South Cass Avenue, Argonne, IL 60439-4844.
E-mail {bhowmick, hovland}@mcs.anl.gov

1 Introduction

Determining an efficient coloring for a given graph has many applications including the evaluation of sparse Jacobian matrices [1]. However, the graph coloring problem is NP-hard [3] and a large volume of literature exists on efficient graph coloring heuristics. In this paper, we present a *backtracking correction algorithm* that can be used in conjunction with a top level heuristic. Use of this algorithm can further decrease the number of colors by dynamically rearranging the colors assigned by the top level heuristic.

2 Backtracking Correction Algorithm

The number of colors used by a greedy graph coloring algorithm is to a large extent determined by the order in which the vertices are colored. The backtracking method attempts to partially change the effects of the underlying vertex ordering by rearranging the mapping of the colors.

The algorithm is invoked whenever a user-specified threshold is exceeded. This threshold should roughly correspond to a lower bound on the expected number of colors. At this point only one vertex (designated as the last vertex) is colored higher than the threshold. The set of colors used by the neighbors of the last vertex gives the acceptable set of colors, i.e. the total number of colors allowed until now.

The last vertex is temporarily assigned a pseudo-color from the acceptable color set. The backtracking algorithm tries to determine whether there is an alternate assignment of colors, not higher than the threshold, to the neighboring vertices that would allow the last vertex to retain the pseudo-color and prevent conflicts. If such an assignment is found, then we have a coloring within the limits of the threshold. The entire list of acceptable colors is searched until such an arrangement is found. If no such arrangement can be obtained, the last vertex is assigned its original color and the threshold is increased by one.

3 Results

Our preliminary experiments are based on sequential distance-1 coloring of adjacency graphs, corresponding to matrices arising in molecular dynamics. The results demonstrate that use of the backtracking algorithm can often succeed in lowering the number of colors used. We experimented with the following ordering heuristics; Natural(N), Largest First(LF), Smallest Last(SL), Incidence Degree(ID), Saturation Degree (SD) [4] and Depth First Search(DFS). For each heuristic we

* This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Offi ce of Computational Technology Research, U.S. Department of Energy under Contract W-31-109-Eng-38.

conducted three sets of experiments using: i) only the heuristic, ii) the heuristic with backtracking and iii) Culberson's Greedy Iterative method [2], with the current heuristic in the first iteration. The results given in Figure 1 for six matrices show that use of the backtracking algorithm almost always gives the smallest number of colors. A comparison of the time taken by the backtracking and Culberson's algorithm (for Saturation Degree) in Figure 2 shows that for sparse matrices the time taken by the backtracking algorithm is lower; however the time increases as the matrix gets denser.

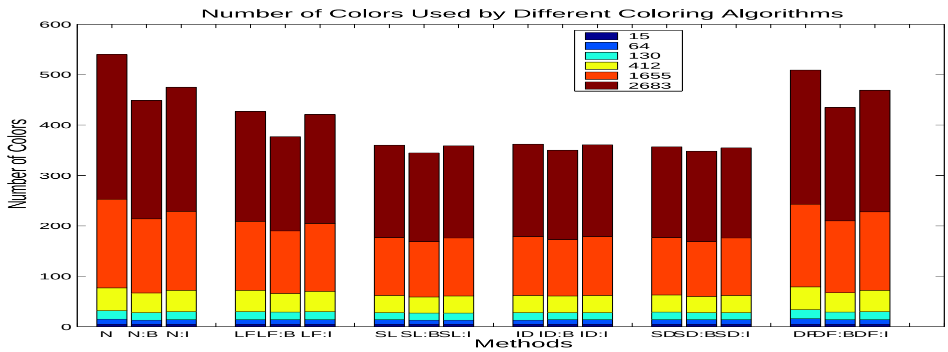

Fig. 1. Performance of different coloring algorithms on a matrix of 4114 vertices. Each colored patch represents different number of edges in the matrix. The edges are given in the order of thousands. From left to right of each set are the heuristic, the heuristic with backtracking and the corresponding Culberson's algorithm.

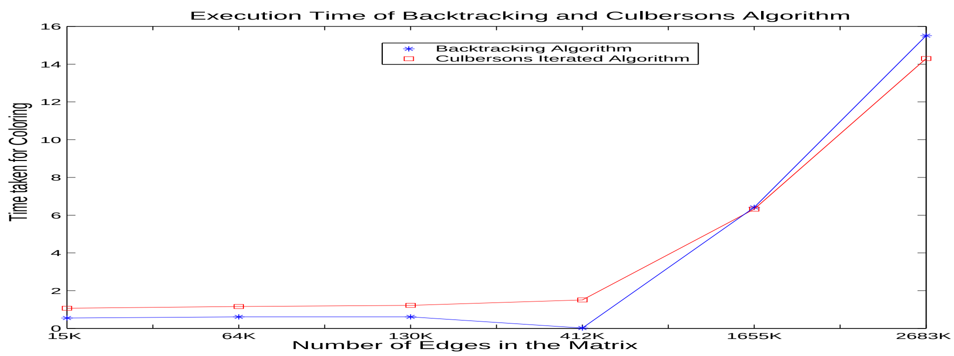

Fig. 2. Comparison of the time taken between Backtracking and Culberson's Algorithm for Saturation Degree; Time is given in seconds.

4 Conclusions

The backtracking algorithm is generally successful in reducing the number of colors. The time taken is competitive with the Culberson's algorithm, in the case of sparse matrices. However, the algorithm is only as good as the top-level heuristic. Future research includes reducing the execution time and minimizing the number of colors.

Acknowledgments The idea for the backtracking algorithm was inspired by reading a preprint of the excellent review on graph coloring algorithms by Assefaw Gebremedhin, Fredrik Manne and Alex Pothen. We are also grateful to Assefaw Gebremedhin and Rahmi Aksu for letting us use their graph coloring software. Our implementation of the backtracking algorithm was built on top of this software.

References

1. T.F. Coleman and J. J. Moré. Estimation of sparse Jacobian matrices and graph coloring problems. *SIAM Journal of Numerical Analysis*, 20(1):187–209, 1983.
2. Joseph C. Culberson. Iterated Greedy Graph Coloring and the Difficulty Landscape. *Technical Report*, 1992.
3. M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freman and Company, New York, 1979.
4. A. Gebremdhin, F. Manne, and A. Pothen. What color is your Jacobian? Graph coloring for computing derivatives. *accepted by SIAM Review*, Dec 2004.