Ordering for Coloring and More

Assefaw H. Gebremedhin * Arijit Tarafdar † Duc Nguyen Alex Pothen

Extended Abstract

Consider the distance-1 coloring problem: given a graph, assign positive integers (called colors) to vertices such that every pair of adjacent vertices receives distinct colors and the number of colors used is minimized. A *greedy* heuristic for this problem progressively extends a partial coloring by processing one vertex at a time, in some order, in each step permanently assigning a vertex the *smallest* allowable color. The order in which the vertices are processed determines the number of colors used by the algorithm. Three ordering techniques that are known to be fairly effective in practice are *Largest First*, *Incidence Degree*, and *Smallest Last*.

The definition typically found in the literature for each of these techniques is intertwined with the coloring algorithm itself. We characterize these in a manner that is independent of a coloring algorithm, which aids the creation of more modular implementations and makes it easier to find other contexts in which the orderings might be useful. The characterization simply uses the notions of back and forward degree. In an ordering $\pi = \{v_1, v_2, \ldots, v_n\}$ of the vertices of a graph G, the *back degree* of the vertex v_i is the number of neighbors of v_i ordered *before* v_i in π, and the *forward degree* of v_i is the number of neighbors of v_i ordered *after* v_i in π. Clearly, the *degree* of v_i is the sum of its back degree and forward degree. Our characterization is summarized in Table 1. The third item in the table, *Dynamic Largest First*, is a new ordering technique we developed and found to be especially effective for a specialized coloring problem to be discussed later.

The ordering variants LF, ID, and DLF are defined left-to-right, from v_1 to v_n. The ith vertex in each case is a vertex with the *largest* degree, back degree, and forward degree, respectively, among the vertices that are yet to be ordered. The heuristic intuition here is that in the course of a greedy algorithm, vertices that are more constrained in the choice of colors get colored early, and thus the algorithm would use fewer colors compared to, say, an arbitrary order. Among the three, ID uses the most accurate measure of "constraint".

An SL ordering is defined right-to-left, from v_n to v_1, and the ith vertex is a vertex with the *smallest back degree* among vertices that are yet to be ordered. In comparison with LF, ID and DLF, an SL ordering aims at minimizing a mathematically better defined objective. Let $B_\pi(G)$ denote the *maximum back degree* in a vertex ordering π. It is easy to see that a greedy algorithm that uses the ordering π to color G will use at most $B_\pi(G) + 1$ colors. Different vertex orderings could result in different maximum back degrees. An SL-ordering gives the *minimum* maximum back-degree over *all* possible orderings, i.e. $B_{SL} = B^* = \min_\pi B_\pi(G)$, where the minimum is taken

*Department of Computer Science and Computing Research Institute, Purdue University, West Lafayette, IN, {agebreme, dnguyen, apothen}@purdue.edu

†Was involved in part of this work while he was a student at Old Dominion University. Currently at Microsoft Corporation, Redmond, WA, aritar@gmail.com

Ordering	Property
Largest First	for $i = 1$ to n:
	v_i has *largest degree* in $V \setminus \{v_1, v_2, \ldots v_{i-1}\}$
Incidence Degree	for $i = 1$ to n:
	v_i has *largest back degree* in $V \setminus \{v_1, v_2, \ldots, v_{i-1}\}$
Dynamic LF	for $i = 1$ to n:
	v_i has *largest forward degree* in $V \setminus \{v_1, v_2, \ldots, v_{i-1}\}$
Smallest Last	for $i = n$ to 1:
	v_i has *smallest back degree* in $V \setminus \{v_n, v_{n-1}, \ldots, v_{i+1}\}$

Table 1: Properties of degree-based ordering techniques, restricted to distance-1 coloring.

	1d partition	2d partition	
Jacobian	*distance-2 coloring*	*star bicoloring*	Direct
Hessian	*star coloring*	NA	Direct
Jacobian	NA	*acyclic bicoloring*	Substitution
Hessian	*acyclic coloring*	NA	Substitution

Table 2: Overview of graph coloring models in computation of derivative matrices. The Jacobian is represented by its bipartite graph, and the Hessian by its adjacency graph. NA stands for not applicable.

over the $n!$ possible vertex orderings. The quantity $B_{SL}(G) + 1$ is a graph property appropriately known as the *coloring number* $col(G)$ of G. We will discuss the interesting connection the parameter $col(G)$ has with several other graph concepts, including degeneracy, core, and arboricity.

In addition to their amenability for description in terms of merely back and forward degrees, the ordering techniques listed in Table 1 are also attractive for their time complexities. Each of them can be implemented to run in linear time in the number of edges in a graph; this is not true for ordering techniques such as Saturation Degree. We have included such efficient implementations in `ColPack`, our software package for graph coloring and related problems in sparse derivative computation.

Besides distance-1 coloring, `ColPack` contains implementations of efficient algorithms for a variety of graph coloring problems in derivative computation. Table 2 gives an overview of these problems and the computational scenarios under which they arise. We have adapted the ordering techniques listed in Table 1 to suit these specialized coloring problems. The adaptation involves extending the notion of degree, since the algorithms involve visits to the distance-2 neighborhood of a vertex in a graph or the coloring is performed on a bipartite graph. In `ColPack`, each ordering technique adapted for the specialized coloring problem is implemented in such a way that its time complexity is upper-bounded by the complexity of the relevant coloring algorithm.

We will present experimental results that compare the ordering techniques—both for the distance-1 coloring and the specialized coloring cases—against each other (and against other orderings such as natural and random) in terms of number of colors used and runtime. The experiments are conducted on large graphs drawn from various application areas as well as on synthetically generated graphs representing different graph classes. The experiments reveal, among other things, that a DLF ordering is especially effective for star bicoloring of some matrix structures and that an SL ordering enables a greedy algorithm to distance-1 color any planar graph using at most 6 colors.