CALCULATION OF CHROMATIC POLYNOMIALS

Larry Basenspiler
Division of Computer and Information Sciences
University of South Alabama

An algorithm called M-Chrom for calculating the chromatic polynomials of sparse graphs is presented. Because of space constraints, the previously developed by Ronald Read algorithm (R-Chrom) handles very sparse graphs on up to 15 vertices, and graphs of medium density on up to 10 vertices. M-Chrom is able to process graphs of any density on up to 32 vertices. M-Chrom is a modified version of R-Chrom and uses stacks instead of queues as the primary data structure. This allows using an amount of space, independent of the density of an input graphs. Using unicyclic target graphs provided a significant speedup when ran for very sparse graph. M-Chrom has been implemented (and R-Chrom made suitable) for running on a vector processor (a Cray X-MP/24) as a vectorized FORTRAN implementation of M-Chrom . Results of experiments demonstrate the greater efficiency of M-Chrom. M-Chrom uses always 10,000 words. Both algorithms, due to the choice of the graph representation, use levels of indirection in referencing the edge list which inhibits vectorization. The vectorization speedup for sparse graphs was limited to two since the share of the most time consuming copy and contract operations is quite large in the execution of the algorithm. In contrast, for mid-dense graphs, vectorization gave a greater speedup. This is due to a greater share of copying in the execution of M-Chrom for dense graphs. Copying is particularly amenable to vectorization, thus the impact of the latter is especially visible. Also, the chord lists of dense graphs are significantly longer than that of sparse graphs, thus the length of loops involved is often large enough to overcome the start-up delays.