A Dynamic Programming Approach for Testing Clique Algorithms

Wendy Myrvold
University of Victoria *

Tania Prsa
IBM, Toronto

Neil Walker
University of Victoria

January 9, 2014

Abstract

Traditionally, practical clique algorithms have been compared based on their performance on various random graphs. We propose a new testing methodology which permits testing to be completed in a fraction of the time required by previous methods.

1 Introduction

An *undirected graph* G consists of a set V of *vertices* and a set E of unordered pairs of vertices called *edges*. The number of vertices is called the *order* of

*Supported by NSERC.

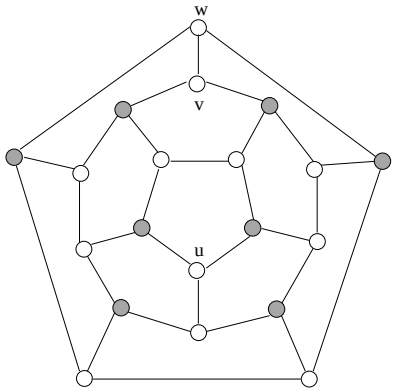

Figure 1: A dodecahedron, showing a maximum independent set (shaded vertices) and also the two pairs, up to isomorphism, of vertices at distances 4 or more, i.e., u, v and u, w.

the graph, and the number of edges is the *size*. Given a graph $G = (V, E)$, a *clique* in the graph G is a set of pairwise adjacent vertices. A *maximum clique* in G is the largest such set of vertices. Although the maximum clique problem is NP-complete, significant progress has been made in developing effective algorithms.

The class of clique algorithms for which our methodology applies are what we call *neighbourhood search algorithms*, and we define these in Section 2. Note how I used a name neigh_search for the section number. The number will be correctly inserted and it means you do not have to worry about renumering when extra sections are inserted in front.

Here is a figure from another paper just to show you how to insert them. The figure is Figure 1. For correct figure numbering, ensure that the label statement below goes after the caption statement.

2 Neighbourhood Search Algorithms

Theorems and lemmas have consecutive labels as defined in the header since it is very confusing for example to have a theorem 1 and also a lemma 1. Here are some examples of Theorem 2.1 and Lemma 2.2.

Theorem 2.1 *If G is a planar graph on three or more vertices then the number of edges is at most $3n - 6$.*

Proof. I made my own commands for starting and ending proofs.

Corollary 2.2 *Planar graphs have at most $O(n)$ edges.*

Proof. The number of edges is less than $3n$ and this is in $O(n)$.

Lemma 2.3 *Lemmas are usually lesser results than a theorem.*

The keywords in these references match what I have in the tan.bib file. One of the earliest approaches, Algorithm CACM 457 [3], searches all vertices except those vertices v where $N[v]$ is completely contained within the neighbourhood of a previously searched vertex. Balas and Yu [2] find a MTIS (Maximally Triangulated Induced Subgraph) as an aid to determine a nice colouring of the vertices in a strategy similar to GREEDY which we describe in the next section. Balas and Xue [1] and also Mannino and Sassano [5] apply fractional colouring to attempt to decrease the number of vertices whose neighbourhoods are searched. Carraghan and Pardalos [4] suggest a simple approach where induced sorting (defined later) is applied to the vertices. Finally, two algorithms [7, 6] are of interest because they provide upper bounds on the time required to apply this recursive strategy.

References

[1] E. Balas and J. Xue. Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring. *Second DIMACS Challenge, CLIQUE Papers*, Oct. 1993. 12 pages.

[2] E. Balas and C. S. Yu. Finding a maximum clique in an arbitrary graph. *SIAM J. Comput.*, 15(4):1054–1068, 1986.

[3] C. Bron and J. Kerbosch. Algorithm 457: finding all cliques of an undirected graph H. *Comm. ACM*, 16(9):575–577, 1973.

[4] R. Carraghan and P.M. Pardalos. An exact algorithm for the maximum clique problem. *Op. Res. Letters*, 9:375–382, 1990.

[5] C. Mannino and A. Sassano. Edge projection and the maximum cardinality stable set problem. Preprint, 1994.

[6] J.M. Robson. Algorithms for maximum independent sets. *J. Alg.*, 7:425–440, 1986.

[7] R.E. Tarjan and A.E. Trojanowski. Finding a maximum independent set. *SIAM J. Comput.*, 6(3):537–546, 1977.