Computers & Operations Research 64 (2015) 293–303

Contents lists available at ScienceDirect

Computers & Operations Research

journal homepage: www.elsevier.com/locate/caor

Infra-chromatic bound for exact maximum clique search

Pablo San Segundo [a,*], Alexey Nikolaev [b], Mikhail Batsyn [b]

[a] Centro de Automática y Robótica (CAR), UPM-CSIC, C/ Jose Gutiérrez Abascal, 2, 28006 Madrid, Spain
[b] Laboratory of Algorithms and Technologies for Networks Analysis, National Research University Higher School of Economics, 136 Rodionova Street, Niznhy Novgorod, Russia

ARTICLE INFO

Available online 24 June 2015

Keywords:
Approximate coloring
Branch-and-bound
MaxSAT
Combinatorial optimization

ABSTRACT

Many efficient exact branch and bound maximum clique solvers use approximate coloring to compute an upper bound on the clique number for every subproblem. This technique reasonably promises tight bounds on average, but never tighter than the chromatic number of the graph.

Li and Quan, 2010, AAAI Conference, p. 128–133 describe a way to compute even tighter bounds by reducing each colored subproblem to maximum satisfiability problem (MaxSAT). Moreover they show empirically that the new bounds obtained may be lower than the chromatic number.

Based on this idea this paper shows an efficient way to compute related "infra-chromatic" upper bounds without an explicit MaxSAT encoding. The reported results show some of the best times for a stand-alone computer over a number of instances from standard benchmarks.

1. Introduction

For a given graph, a complete subgraph, alias clique, is a graph which vertices are all pairwise adjacent. Finding a hidden clique with the maximum number of vertices is a deeply studied NP-hard problem known as the *maximum clique problem* (MCP). MCP has found many applications in a wide scope of fields [1]. Correspondence related problems appear in computational biology, [2], robotics [3–4], computer vision [5]. Finding cohesive clusters in networks is another typical application. Its use in mining correlated stocks is also worth noting.

Exhaustive clique enumeration behind exact MCP can be traced back to the Bron and Kerbosch algorithm [6]. A basic branch and bound (BnB) algorithm which computes primitive bounds for every subproblem was described in [7]. Since then, researchers have tried to devise ways of establishing tighter bounds. Fahle [8] and Régin [9] use a constraint-based approach to prune the search space, but the majority of current efficient solvers are color-based, i.e. they employ a greedy coloring heuristic to compute an upper bound for the clique number of every subproblem, as in [10–19]. Interesting recent comparison surveys for exact maximum clique algorithms have been reported by Prosser [20] and Wu and Hao's [24]. They show MCS [15], MaxCLQ [18], and bit optimized BBMC [12–13] as the current fastest algorithms at present.

The theoretical foundation for approximate color-based algorithms can be found in the following proposition [21]: *the chromatic number of every graph G is an upper bound on its clique number*

$$\chi(G) \geq \omega(G) \quad (1)$$

It is worth noting that, although color-based bounds are reasonably tight on average, Mycielski showed how to build graphs in which $\chi(G) - \omega(G) \geq n, \ \forall n \in \mathbb{N}$ [22]. This constitutes a drawback for the color-based approach.

Li and Quan in [17,18] show empirically that it is possible to compute better approximations to the clique number than the chromatic number by encoding a colored graph to MaxSAT and apply typical logical inferences.

This paper describes a new efficient upper bound related to the previous bound but reasoning with color set information. The procedure is applicable to a more constrained set of cases than [17,18] but is also faster to compute since it does not require excessive MaxSAT inferences.

1.1. Preliminaries

A simple undirected graph $G = (V, E)$ consists of a finite set of vertices $V = \{v_1, v_2, \cdots, v_n\}$ and edges $E \subseteq V \times V$ that pair distinct vertices. Two vertices are said to be adjacent (neighbors) if they are connected by an edge. For any vertex $v \in V$, $N_G(v)$ (or simply $N(v)$ when the graph is clear from the context) refers to the neighbor set of v in G. Any subset of vertices $U \subseteq V$ *induces* a new subgraph $G = (U, E') = G[U]$ with vertex set U and edge set $E' \subseteq E$ such that

* Corresponding author.
Tel.: +34 91 7454660, +34 91 3363061; fax: +34 91 3363010.
E-mail address: pablo.sansegundo@upm.es (P. San Segundo).

http://dx.doi.org/10.1016/j.cor.2015.06.009
0305-0548/

both endpoints of any edge in E' are in U. $G_v \subseteq G$ denotes the graph induced by the neighbor set of vertex v in G.

Some definitions used in this paper are:

- *maximal clique*: a clique that cannot be enlarged by any other vertex in the graph;
- *maximum clique*: a clique which has maximum order (number of vertices);
- *independent set*: a subset of a graph G, which elements are pairwise non-adjacent;
- *vertex coloring*: an assignment of colors $c(v): V \to \mathbb{N}$ to every vertex of a graph so that any two adjacent vertices have different colors;
- *chromatic number* $\chi(G)$: the minimum number of colors in which it is possible to color graph G. Finding the chromatic number of a graph is an NP-hard problem;
- *clique number* $\omega(G)$: the number of vertices of a maximum clique in G;
- *sequential vertex coloring*: an approximate coloring procedure that iteratively colors vertices in some predefined order;
- *greedy vertex coloring* (SEQ): a sequential vertex coloring in which every vertex is assigned the smallest possible color.

Additional standard notation in the paper is $\deg(v)$for vertex v degree and ΔG for the maximum degree of a graph G. Color set notation $C(G) = \{C_1, C_2, \cdots, C_k\}$denotes a vertex coloring of size k, i.e. a coloring that employs k different color numbers. $C(G)$ partitions the vertex set into k disjoint independent color sets C_i, each one containing every vertex with color number i.

2. The branch-and-bound algorithm

Most efficient BnB algorithms for the MCP employ some form of systematic enumeration of maximal cliques together with SEQ heuristic to compute an upper bound for the clique number in every subproblem.

Although other formulations are possible (e.g. a binary tree), clique enumeration is typically described as a recursive procedure that branches on every candidate vertex in order to enlarge the clique in the current node. This leads to a binomial search tree in which, at depth level k, every possible clique of size k is considered [7]. Leaf nodes hold maximal cliques in the path to the root node and are evaluated to test whether their clique number is greater than the best clique found so far. If this is true, they replace the existing solution.

Let S be the clique at the current node and let S_{max} be the best solution found during search at any moment. All enumerations starting from S that cannot improve S_{max} may be pruned. For any candidate vertex v, color-based BnB solvers use $c(v)$ as an upper bound for the clique number of the hanging subproblem G_v. The pruning condition is typically formulated as:

$$|S| + c(v) \le |S_{max}| \quad (2)$$

Konc and Janečič in [10] made an interesting interpretation of (2). They define parameter $k_{min} = |S_{max}| - |S| + 1$, so that the pruning condition (2) becomes $c(v) < k_{min}$. Leading algorithms use k_{min} explicitly to advantage during the bounding phase, as in the paper.

Tomita and Seki in [14] introduced the idea of branching on maximum color. This has two important consequences: 1. If any candidate vertex in $C_{k_{min}-1}$ is pruned, then all the remaining vertices are also pruned; 2. Search is directed towards large cliques (a coloring of a clique of size k must have size k).

Candidate vertices with the smallest degree should be picked first at the root node to reduce the size of the search tree. A common strategy for initial vertex sorting is the following heuristic used to compute graph degeneracy: at the beginning, a vertex v with minimum degree is removed from the initial set V and placed last in the new list V'. In the next iteration, the vertex with minimum degree in graph $G \setminus \{v\}$ is placed one before last in V'; the process continues until all vertices are ordered. At the root node vertices should be selected by non-decreasing degree to reduce branching, so they are picked in *reverse order* from V'. It is worth noting that a number of more sophisticated orderings concerning tie-breaks have been suggested in practice (see e.g. [15]).

A simple upper bound for every vertex at the root node is the minimum value between its position in the ordering and the maximum degree of the graph plus 1. It is used in leading solvers BBMC and MCS and others. Tighter bounds may possibly produce smaller search trees as suggested in [19].

Another important idea also used in MCS and BBMC is to keep the initial vertex ordering fixed throughout the search. This produces on average tighter SEQ colorings compared to vertices sorted according to color as proposed in earlier algorithms [10,14]. Besides this *implicit branching strategy*, MCS also describes *recoloring* (Re-NUMBER is the term given to the procedure in the original paper). Recoloring is a repair mechanism which is also related to infra-chromatic bounding described in this paper. It is discussed in Section 4 inside the proposed new algorithmic framework.

Procedures described in the paper employ the following variables:

- U: a list of candidate vertices sorted according to the initial ordering;
- U_v: a list of candidate vertices in the child subproblem which contains only neighbors of vertex v;
- S: the current clique;
- S_{max}: the best clique found so far;
- L: a list of candidate vertices sorted according to color (highest color last);
- L_v: a list of candidate vertices sorted according to color in the child subproblem which contains only neighbors of vertex v;
- F: a list of forbidden colors;
- $c(v)$: the color number assigned to vertex v;
- $N_U(v)$: the neighbor set of vertex v from the list of candidate vertices in U;
- k_{min}: a pruning threshold, i.e. vertices assigned a lower color number will be pruned;
- $C(G_v)$: SEQ coloring of G_v, the subgraph induced by v;
- C_k: a color set of a SEQ coloring that contains all vertices with color number k.

Listing 1 outlines the reference MCP algorithm described previously. Steps 1–2, 4–10, 13–15 implement systematic enumeration (with repetitions). Step 3 is the pruning step, which depends on the upper bounds computed in UPPERBOUND (step 12). UPPERBOUND calls the new bounding procedure described in Section 5.

Listing 1. Outline of the reference MCP algorithm

Input: a simple graph $G = (V, E)$with vertices sorted by *smallest degree-last*
Output: a maximum clique with vertex set S_{max}
REFMC (U, S, S_{max}, C, L)
Initial step:
$U \leftarrow V$, $S \leftarrow \phi$, $S_{max} \leftarrow \phi$, $c(v_i) := \min\{i, \Delta G + 1\}$, $L \leftarrow V$
1. **repeat until** $U = \phi$
2. select a vertex v from L in reverse order //maximum color branching
3. **if** ($|S| + c(v) \le |S_{max}|$) **then return** //pruning step