Annales Univ. Sci. Budapest., Sect. Comp. 48 (2018) 199–208

A CLIQUE SIZE ESTIMATE BASED ON COLORING THE NODES OF CERTAIN SUBGRAPHS

Sándor Szabó (Pécs, Hungary)

Communicated by Ferenc Schipp

(Received February 7, 2018; accepted June 26, 2018)

Abstract. Many of the maximum clique search algorithms used in practice employ a routine to establish upper estimate of the clique size of a given graph. The upper estimates are typically based on legally coloring the nodes in some greedy manner. Motivated by these facts we propose a upper estimate for the clique number. Using any greedy coloring procedure the nodes of many subgraphs are colored legally and then these partial results are combined together to a clique size upper estimate.

1. Introduction

Let $G = (V, E)$ be a finite simple graph. This means that G has finitely many nodes and finitely many edges. Further G does not have any loops or double edges. Here V is the set of nodes of the graph. In this situation the set of edges E can be identified with a set unordered pairs of the elements of V. A subgraph Δ of G is called a clique in G if each two distinct nodes in Δ are always connected by an edge in G. The number of the nodes of Δ is the size of the clique. For each finite simple graph G there is a well defined number k such that G contains a clique of size k but G does not contain any clique of size $k + 1$. This number k is called the clique number of G and it is denoted by $\omega(G)$. The problem of computing the clique number of a graph is an optimization problem. It is well known that this optimization problem is NP hard. (See [3].)

Key words and phrases: Clique number, chromatic number, legal coloring of the nodes.
2010 Mathematics Subject Classification: Primary 05C15; Secondary 05B45, 52C22.

We color the nodes of G such that each node of G receives exactly one color and two distinct nodes that are connected by an edge never receive the same color. This type of coloring of the nodes is called legal coloring. For each finite simple graph G there is well defined number k such that the nodes of G can be legally colored using k colors but the nodes of G cannot be legally colored using $k-1$ colors. This number k is called the chromatic number of G and it is denoted by $\chi(G)$. The problem of computing $\chi(G)$ is an optimization problem. This optimization problem belongs to the NP hard complexity class. (See [3].) There is a large variety of greedy algorithms for coloring the nodes of a given graph legally. (See [2], [4].) These algorithms typically have polynomial running times but do not provide the optimum number of colors. They only give an upper bound of the chromatic number. As the inequality $\omega(G) \leq \chi(G)$ obviously holds the greedy coloring procedures can be used to estimate clique number.

In order to reduce the search space a number of practical clique search algorithms employ legal node colorings. The role of the coloring is to find an upper estimate of the clique size. (See [10], [9], [8].) This is the main motivation of the result of this short note.

The procedure we propose to estimate the clique is fairly flexible. At a certain point it may incorporate any available clique size estimating algorithm. Beside legal coloring of the nodes there are less elementary methods to estimate the clique size of a graph. They typically give better upper bound for the clique number on the expense of more computations. As an example we mention the graph parameter $\theta(G)$ defined by L. Lovász. It has the property that $\omega(G) \leq \theta(G) \leq \chi(G)$ and $\theta(G)$ can be computed in polynomial time by solving a semi-definite program.

2. Description of the estimating procedure

Let $G = (V, E)$ be a finite simple graph. We divide the set V into two disjoint subsets U and W. There is a large number of ways one can choose U and V. The choice of these subsets influences the clique size estimate. We postpone the issue of assessing the advantages of the various choices. At this moment we simply assume that there is method of picking the subsets U and W.

Let H be the subgraph of G induced by U and let K be the subgraph of G induced by W. Using a greedy algorithm we construct a legal coloring of the nodes of the graphs H and K. Let $C_1, \ldots, C_p$ be the colors classes of H and let $D_1, \ldots, D_q$ be the colors classes of K.

For each node v of G we define a quantity which will be called the clique degree of v and will be denoted by $\mathrm{cldeg}(v)$. Suppose first that $v \in U$. Let us consider the subgraph L_v of G induced by $N(v) \cap W$. Here $N(v)$ is the set of nodes in G that are adjacent to v. The set $N(v) \cap W$ consists of the neighbors of v restricted to the set W. Using a fixed algorithm we establish an upper estimate of $\omega(L_v)$. This well defined number will be $\mathrm{cldeg}(v)$. For the sake of definiteness we may use a greedy node coloring procedure to establish an upper bound for $\omega(L_v)$. However, as the graph L_v is smaller than the graph G we may use computationally more expensive procedures to get better estimates. In addition computing the clique degrees for different nodes can be carried out independently of each other.

When $v \in W$, then subgraph L_v is induced by $N(v) \cap U$ instead of $N(v) \cap W$.

After this phase of the algorithm is completed the clique degree of each node of G is available. We may turn to constructing the profiles of the graph H and K. Set

$$\alpha_i = \max\{\mathrm{cldeg}(v) : \ v \in C_i\},$$

for each i, $1 \leq i \leq p$. We order the numbers $\alpha_1, \ldots, \alpha_p$ into a non-increasing list $\alpha'_1, \ldots, \alpha'_p$ and we call this list the profile of the graph H. In a similar way set

$$\beta_j = \max\{\mathrm{cldeg}(v) : \ v \in D_j\},$$

for each j, $1 \leq j \leq q$. We order the numbers $\beta_1, \ldots, \beta_q$ into a non-increasing list $\beta'_1, \ldots, \beta'_q$ and we call this list the profile of the graph K.

After completing this phase of the procedure the profiles of the graphs H and K are completed. We consider the ordered pairs (r, s), $0 \leq r \leq p$, $0 \leq s \leq q$. We say that the ordered pair (r, s) is not a qualifying pair if at least one of the following inequalities holds.

$$\alpha'_1 < s, \ldots, \alpha'_r < s \tag{2.1}$$

$$\beta'_1 < r, \ldots, \beta'_s < r \tag{2.2}$$

In other words the pair (r, s) is qualifying if each of the following inequalities holds.

$$\alpha'_1 \geq s, \ldots, \alpha'_r \geq s \tag{2.3}$$

$$\beta'_1 \geq r, \ldots, \beta'_s \geq r \tag{2.4}$$

Note that in the $r = 0$ case each of the inequalities in (2.3) holds. Similarly in the $s = 0$ case each of the inequalities in (2.4) holds. The number of the pairs

(r, s) is equal to $(p+1)(q+1)$. An inspection can be used to find the qualifying pairs. Set $t = \max\{r + s : (r, s) \text{ is qualifying}\}$.

The algorithm gives the $\omega(G) \leq \max\{p, q, t\}$ estimate for the clique size of G. A straight-forward way to compute t is to arrange the (r, s) pairs into the following list

$$(p, q), (p-1, q), (p, q-1), (p-2, q), (p-1, q-1), (p, q-2), \ldots$$

and go on this list until we locate the first qualifying pair.

3. Justification of the procedure

We will use the notations of the previous sections and prove the following result.

Lemma 3.1. *Let $G = (V, E)$ be a finite simple graph and assume that G has at least one node. The quantity $t = \max\{r + s : (r, s) \text{ is qualifying}\}$ is an upper bound of $\omega(G)$.*

Proof. The graph G contains a clique Δ of size $\omega(G)$. Let U' be the set of nodes of Δ in U and let W' be the set of nodes of Δ in W. Clearly U' and W' are disjoint sets and $|U'| + |W'| = \omega(G)$.

We distinguish the next four cases.

$$\begin{array}{lll} \text{Case } 1: & U' = \emptyset, & W' = \emptyset. \\ \text{Case } 2: & U' = \emptyset, & W' \neq \emptyset. \\ \text{Case } 3: & U' \neq \emptyset, & W' = \emptyset. \\ \text{Case } 4: & U' \neq \emptyset, & W' \neq \emptyset. \end{array}$$

In case 1 the clique Δ does not have any nodes. As Δ is a maximum clique in G it follows that G does not have any node. By the assumption of the lemma this is not possible.

Let us turn to case 2. Since $U' = \emptyset$ it follows that Δ is a clique in the subgraph K of G induced by W. The nodes of K are legally colored using q colors and so $\omega(G) \leq q$ holds. Note that the ordered pair $(0, q)$ is a qualifying pair. This gives that $q \leq t$. Thus $\omega(G) \leq t$ as required.

Case 3 can be settled in an analogous way. The equation $W' = \emptyset$ implies that Δ is a clique in the subgraph H of G induced by the set of nodes U. The nodes of H are legally colored using p colors. Therefore $\omega(G) \leq p$. As the ordered pair $(p, 0)$ is a qualifying pair we get $p \leq t$ and so $\omega(G) \leq t$.

We are left with case 4. The elements of the set $U' \cup W'$ are the nodes of the clique Δ. Consequently the unordered pair $\{u, w\}$ is an edge of G for each $u \in U'$, $w \in W'$. The subgraph L_u of G induced by $N(u) \cap W$ must contain a clique of size s. There are r choices for the node $u \in U'$. These choices show that the inequalities (2.3) must hold. Similarly, the subgraph L_w of G induced by $N(w) \cap U$ has to contain a clique of size r. There are s choices for the node $w \in W'$. This is why the inequalities (2.4) must hold.

We can see that the ordered pair (r, s) is a qualifying pair. The inequality $\omega(G) \leq r + s$ holds for each qualifying pair (r, s). Thus $\omega(G) \leq t$ as required.

■

4. A toy example

In order to illustrate the estimating procedure we work out an example in details.

Example 4.1. Consider the graph $G = (V, E)$ given by the adjacency matrix in Table 1. The graph has 16 nodes and 56 edges. The nodes are denoted by $1, \ldots, 16$.

We partition the set of nodes of G into two sets. Namely, we set

$$U = \{1, \ldots, 8\}, \quad W = \{9, \ldots, 16\}.$$

Clearly U and W are disjoint and their union is equal to V the set of nodes of G. We define two new graphs. Let H and K be subgraph of G spanned by U and W, respectively.

Using the simplest greedy sequential coloring algorithm we color the nodes of H and K. The color classes of H are

$$C_1 = \{1, 5, 8\},\ C_2 = \{2, 3\},\ C_3 = \{4, 6\},\ C_4 = \{7\}.$$

The color classes of K are

$$D_1 = \{9, 11, 16\},\ D_2 = \{10, 12, 14\},\ D_3 = \{13, 15\}.$$

Let $u \in U$. We define a clique degree of u with respect to W. We consider the set $N(u) \cap W$. This set induces a subgraph L_u in G. We legally color the nodes of L_u and use the number of colors as an upper estimate of the clique degree of u.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
1	×	•	•	•			•		•		•		•		•	
2	•	×			•	•		•				•	•		•	•
3	•		×	•		•	•		•	•		•	•	•		
4	•		•	×	•		•	•					•			
5		•		•	×	•			•			•		•		
6		•	•		•	×			•	•			•		•	
7	•		•	•			×	•			•			•		•
8		•		•			•	×	•		•	•				•
9	•		•		•	•		•	×	•			•		•	
10			•			•			•	×			•	•		
11	•						•	•			×	•			•	
12		•	•		•			•			•	×	•		•	
13	•	•	•	•		•			•	•		•	×	•		•
14			•		•		•			•			•	×		•
15	•	•				•			•		•	•			×	•
16		•					•	•					•	•	•	×

1	2	3	4	7	9	11	13	15		
2	1	5	6	8	12	13	15	16		
3	1	4	6	7	9	10	12	13	14	
4	1	3	5	7	8	13				
5	2	4	6	9	12	14				
6	2	3	5	9	10	13	15			
7	1	3	4	8	11	14	16			
8	2	4	7	9	11	12	16			
9	1	3	5	6	8	10	13	15		
10	3	6	9	13	14					
11	1	7	8	12	15					
12	2	3	5	8	11	13	15			
13	1	2	3	4	6	9	10	12	14	16
14	3	5	7	10	13	16				
15	1	2	6	9	11	12	16			
16	2	7	8	13	14	15				

Table 1. The adjacency matrix and the lists of neighbors of the graph in Example 4.1.

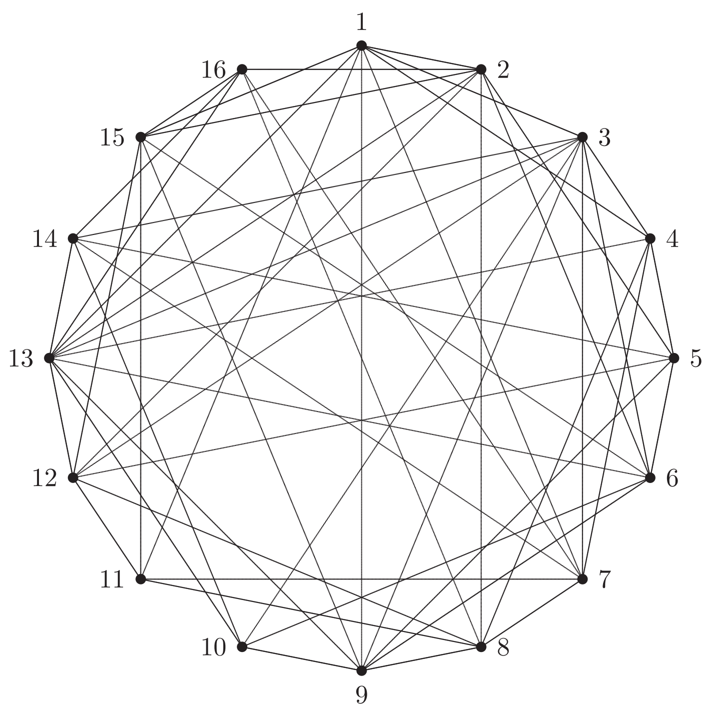

Figure 1. A geometric representation of the graph G in Example 4.1.

We may use any coloring procedure we choose. This means that different people will find different estimates of the clique degrees of the node u. The clique number of L_u is uniquely determined but it might be hard to compute. This is the reason why we are satisfied with a not well defined but efficiently computable estimate of the clique degree.

Let $w \in W$. We define a clique degree of w with respect to U in a similar way. The clique degree estimates of the of the nodes of H and K are given in Table 2.

From each of the color classes C_1, C_2, C_3, C_4 we pick a node with a largest clique degree estimate. We get four number in this way. We then order these numbers in a non-increasing order. We call the resulting sequence the profile of K.

Similarly, from each of the color classes D_1, D_2, D_3 we choose a node with a maximum clique degree estimate. Then we order the three numbers in a non-increasing order. The sequence we get is the profile of K. The profiles of H and K are given in Table 2.

node	1	5	8	2	3	4	6	7
clique degree estimate	2	1	2	2	3	1	3	2

node	9	11	16	10	12	14	13	15
clique degree estimate	3	2	2	3	2	2	3	2

profile of H	3	3	2	2
profile of K	3	3	3	

Table 2. The nodes with color degrees and the color degree profiles of H and K in Example 4.1.

Suppose there is a clique Δ in G such that r nodes of Δ is in U and s nodes of Δ in W. It means that r nodes in U are all adjacent to s nodes in W . We are working with a complete bipartite graph B of type (r, s).

In Table 3. we test first the $r = 4$, $s = 3$ possibility. Note that the clique degree of each node of B in the set U is at least 3. Similarly, the clique degree of each node of B in the set W is at least 4. We can compare the required clique degrees with the available clique degrees. The + signs indicates that $r = 4$, $s = 3$ choice is not possible.

The final conclusion we can read off from Table 3 is that $\omega(G) \leq 5$.

$r = 4, s = 3$							
needed	3	3	3	3	4	4	4
found	3	3	2	2	3	3	3
			+	+	+	+	+
$r = 3, s = 3$							
needed	3	3	3		3	3	3
found	3	3	2		3	3	3
			+				
$r = 4, s = 2$							
needed	2	2	2	2	4	4	
found	3	3	2	2	3	3	
					+	+	
$r = 2, s = 3$							
needed	3	3			2	2	2
found	3	3			3	3	3

Table 3. Testing the (r, s) pairs in Example 4.1.

References

[1] **Balas, E. and J. Xue,** Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring, *Algorithmica* **15** (1996), 397–412.

[2] **Brelaz, D.,** New methods to color the vertices of a graph, *Communications of the ACM* **22** (1979), 251–256.

[3] **Garey, M. R. and D. S. Johnson,** *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, New York, 2003.

[4] **Leighton, F. T.,** A graph coloring algorithm for large scheduling problems, *Journal of Research of National Bureau of Standards* **84** (1979), 489–506.

[5] **Östergård, P. R. J.,** A fast algorithm for the maximum clique problem, *Discrete Applied Mathematics* Ser. A **120** (2002), 197–207.

[6] **Carraghan, R. and P. M. Pardalos,** An exact algorithm for the maximum clique problem, *Operations Research Letters*, **9** (1990), 375–382.

[7] **Prosser, P.,** Exact algorithms for maximum clique: A computational study, *Algorithms* **5** (2012), 545–587.

[8] **P. San Segundo, P. and C. Tapia,** Relaxed approximate coloring in exact maximum clique search, *Computers and Operations Research.* **44** (2014), 185–192.

[9] **Tomita, E. and T. Kameda,** An efficient branch-and-bound algorithm for finding a maximum clique with computational experiments, *Journal of Global Optimization* **37** (2007), 95–111.

[10] **Wood, D. R.,** An algorithm for finding a maximum clique in a graph, *Operations Research Letters* **21** (1997), 211–217.

S. Szabó
Institute of Mathematics and Informatics
Faculty of Science
University of Pécs
Pécs
Hungary
sszabo7@hotmail.com