

LOWER ESTIMATE OF CLIQUE SIZE VIA EDGE COLORING

Balázs Király^{1,a} and Sándor Szabó^{1,*}

¹ Institute of Mathematics and Informatics, University of Pécs, Ifjúság u. 6, 7624 Pécs, Hungary

^a kiralyb@gamma.ttk.pte.hu

Communicated by László Tóth

Original Research Paper

Received: Sep 15, 2020 • Accepted: Nov 27, 2020

First published online: Mar 24, 2021

© 2020 The Author(s)



ABSTRACT

In many clique search algorithms well coloring of the nodes is employed to find an upper bound of the clique number of the given graph. In an earlier work a non-traditional edge coloring scheme was proposed to get upper bounds that are typically better than the one provided by the well coloring of the nodes. In this paper we will show that the same scheme for well coloring of the edges can be used to find lower bounds for the clique number of the given graph. In order to assess the performance of the procedure we carried out numerical experiments.

KEYWORDS

clique number, chromatic number, maximum clique, greedy coloring, clique size estimates

MATHEMATICS SUBJECT CLASSIFICATION (2020)

Primary 05C15; Secondary 05B45, 52C22

1. INTRODUCTION

The graphs in this paper are always finite graphs, that is, they have finitely many nodes and finitely many edges. The graphs are without loops and double edges, that is, they are simple graphs. Let $G = (V, E)$ be a finite simple graph, where V is the set of nodes and E is the set of edges of G .

A subgraph Δ of G is called a clique in G if two distinct nodes of Δ are always adjacent in G . A clique with k nodes will be called a k -clique. A node of the graph can be viewed as a 1-clique and an edge can be viewed as a 2-clique. For each finite simple graph G there is an integer k such that it contains a k -clique but it does not contain any $(k + 1)$ -clique. This well defined number k is called the clique number of G and it is denoted by $\omega(G)$.

PROBLEM 1.1. Given a finite simple graph $G = (V, E)$. Let us determine $\omega(G)$.

PROBLEM 1.2. Given a finite simple graph $G = (V, E)$ and given a positive integer k . Let us decide if G contains a k -clique.

Problems 1.1 and 1.2 are referred to as the maximum clique and k -clique problem, respectively. It is well-known that these problem are in the NP-hard complexity class. (For further details see [4].) Both problems have many applications in applied discrete mathematics.

* Corresponding author. E-mail: szabos@gamma.ttk.pte.hu

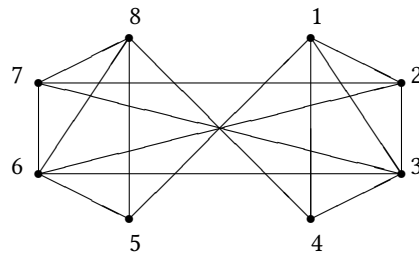


FIGURE 1. A graphical representation of the graph G in Example 3.1.

The typical clique search algorithms used in practice establish upper and lower bounds for the clique number of the given graph. If the lower and upper estimates coincide, then the clique number of the graph is computed. If there is a gap between the upper and lower estimates, then we divide the clique search instance into smaller instances. In short we carry out an optimality test and when this test is inconclusive a branching takes place.

At many combinatorial type algorithms the upper bound for the clique number is coming from a well coloring of the nodes of the graph. In [6] non-conventional edge coloring scheme was proposed to get upper bound for the clique number. The edge coloring based upper bounds are typically sharper but they come for a higher computational cost. In this short note we will show that the same edge coloring scheme can be used to find lower bounds of the clique number of the given graph. We will carry out numerical experiments to see how the proposed procedure works in practical setting.

TABLE 1. The adjacency matrix of the graph in Example 3.1.

	1	2	3	4	5	6	7	8
1	×	•	•	•	•			
2	•	×	•				•	•
3	•	•	×	•			•	•
4	•		•	×				•
5	•				×	•		•
6		•	•		•	×	•	•
7		•	•			•	×	•
8				•	•	•	•	×

2. THE EDGE AUXILIARY GRAPH

Let us assume that we are given a finite simple graph $G = (V, E)$. Using G we construct a new auxiliary graph $\Gamma = (W, F)$. The nodes of Γ are the unordered pairs $w = \{x, y\}$ for which $x, y \in V$, $x \neq y$ and the unordered pair $\{x, y\}$ is an edge of the graph G . Let us consider two distinct nodes

$$w_1 = \{x_1, y_1\} \text{ and } w_2 = \{x_2, y_2\} \tag{2.1}$$

of the graph Γ and set $X = \{x_1, y_1, x_2, y_2\}$. If the subgraph L induced by X in G is a clique in G , then we say the subgraph L is a qualifying subgraph of G . The distinct nodes (2.1) of Γ are adjacent in Γ whenever the associated subgraph L is a qualifying subgraph of G . We call the graph Γ the edge auxiliary graph associated with the graph G .

Note that the subset X has either 3 or 4 elements as the nodes (2.1) are distinct. Consequently, when L is qualifying, then it is either a 3-clique or a 4-clique in G .

LEMMA 2.1. If the graph G contains a k -clique, then the edge auxiliary graph $\Gamma = (W, F)$ contains an s -clique, where $s = k(k - 1)/2$.



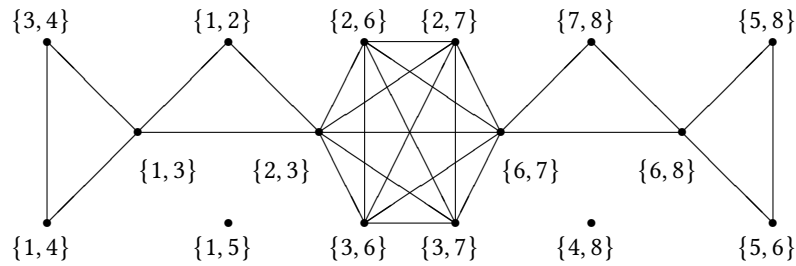


FIGURE 2. A possible geometric representation of the edge auxiliary graph Γ in Example 3.1.

Proof. Suppose that Δ is a k -clique of the graph G and let U be the set of nodes of Δ . Set $T = \{ \{x, y\} : x, y \in U, x \neq y \}$. The set T contains $s = k(k - 1)/2$ elements. If (2.1) are two distinct elements of T , then the subgraph L induced by the subset $X = \{x_1, y_1, x_2, y_2\}$ is a qualifying subgraph of G . Thus the elements of T are the nodes of an s -clique in Γ . \square

Next assume that we have located an s -clique Ω in the edge auxiliary graph Γ and $T = \{ \{x_1, y_1\}, \dots, \{x_s, y_s\} \}$ is the set of nodes of Ω . Let u_1, \dots, u_k be all the distinct elements appearing on the list $x_1, y_1, \dots, x_s, y_s$.

LEMMA 2.2. In the situation described above the elements u_1, \dots, u_k are the nodes of a k -clique Δ in the graph G .

Proof. Let us consider the following list of ordered pairs

$$\begin{matrix} (u_1, u_1) & \dots & (u_1, u_k) \\ \vdots & \ddots & \vdots \\ (u_k, u_1) & \dots & (u_k, u_k) \end{matrix} \tag{2.2}$$

arranged into k rows and k columns. We underline the ordered pairs (u_i, u_j) and (u_j, u_i) whenever the unordered pair $\{u_i, u_j\}$ is an element of the set T .

Clearly, the ordered pairs in the main diagonal are not underlined. Further if the ordered pair (u_i, u_j) is underlined, then so is the ordered pair (u_j, u_i) . Note that for each index $i, 1 \leq i \leq k$ the element u_i must appear in an unordered pair $\{x_t, y_t\}$ of the set T for some index $t, 1 \leq t \leq s$. In other words the first row of (2.2) must contain at least one underlined element. In general each row of list (2.2) contains at least one underlined pair. A similar reasoning gives that each column of list (2.2) contains at least one underlined pair.

If the ordered pair (u_i, u_j) on list (2.2) is underlined, then the unordered pair $\{u_i, u_j\}$ is an element of the set T . This means that the unordered pair $\{u_i, u_j\}$ is a node of the clique Ω . In particular, it is an edge of the graph G . It remains to prove that if the unordered pair (u_i, u_j) appears on list (2.2) it is still an edge of the graph G .

In order to do so note that there is an underlined ordered pair (u_i, u_t) on list (2.2) for some index $t, 1 \leq t \leq k$. Similarly, there is an underlined ordered pair (u_s, u_j) on list (2.2) for some index $s, 1 \leq s \leq k$. As the ordered pairs $(u_i, u_t), (u_s, u_j)$ are underlined, the unordered pairs $\{u_i, u_t\}, \{u_s, u_j\}$ are nodes of the clique Ω in the edge auxiliary graph Γ . Thus the subgraph L induced by the set $X = \{u_i, u_t, u_s, u_j\}$ in the graph G is a qualifying subgraph. As L is a clique in G , it follows that the unordered pair $\{u_i, u_j\}$ is an edge of the graph G . \square

In section 2 we have reduced of spotting a splitting partition to spotting a clique in the edge auxiliary graph. There is a large number of algorithms for locating a not necessarily maximum clique in a given graph. In the literature they came under the name of non-exact clique search algorithms. Coloring of the nodes can be used to locate suboptimal cliques in a straight-forward manner.

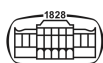


TABLE 2. The adjacency matrix of the edge auxiliary graph Γ in Example 3.1.

	1	1	1	2	2	2	3	3	3	4	5	5	6	6	7		
	2	3	4	5	3	6	7	4	6	7	8	6	8	7	8	8	
1,2	×	•			•												
1,3	•	×	•		•			•									
1,4		•	×					•									
1,5				×													
2,3	•	•			×	•	•		•	•				•			
2,6					•	×	•		•	•				•			
2,7					•	•	×		•	•				•			
3,4	•	•						×									
3,6					•	•	•		×	•				•			
3,7					•	•	•		•	×				•			
4,8												×					
5,6													×	•		•	
5,8													•	×		•	
6,7					•	•	•		•	•				×	•	•	
6,8													•	•	•	×	•
7,8														•	•	×	

We assign colors to the nodes of a given graph G . This assignment of colors that satisfies the following two conditions is called a well coloring of the nodes of G .

- (1) Each vertex receives exactly one color.
- (2) The two end points of an edge cannot receive the same color.

For each finite simple graph G there is a well defined positive integer k such that the nodes of G can be well colored using k colors and the nodes of G cannot be well colored using $k - 1$ colors. This k is called the chromatic number of G and it is denoted by $\chi(G)$.

PROBLEM 2.3. Given a finite simple graph $G = (V, E)$. Determine $\chi(G)$.

PROBLEM 2.4. Given a finite simple graph $G = (V, E)$ and given a positive integer k . Decide if the nodes of G can be well colored using k colors.

Problem 2.4 is known as the k -coloring problem. It is a decision problem and it is a well known result that it belongs to the NP-complete complexity class. The problem of determining the chromatic number of a given graph is an NP-hard problem. (For further details see [4].) It is an empirical fact that well coloring the nodes of G using not necessarily the optimal number of colors has practical utility. In this paper we will use only one approximate coloring algorithms, the simple sequential greedy node coloring algorithm. (For further details see [3], [2], [1].)

A well coloring of the nodes of the finite simple graph $G = (V, E)$ can be conveniently described by a function $f : V \rightarrow \{1, \dots, k\}$. Here the numbers $1, \dots, k$ stand for the colors and the equation $f(v) = i$ expresses the fact that node v receives color i . The set of nodes $C_i = \{v : v \in V, f(v) = i\}$ is called the i -th color class. It is the set of nodes of G colored by color i .

It is plain that a color class is an independent set of the graph G . Therefore the elements of a color class form the nodes of a clique in \overline{G} the complement graph of G . So when we are looking for a clique in the edge auxiliary graph to locate a splitting partition we may do this by well coloring the nodes of the complement of the auxiliary graph. We may pick the elements of any color class as the nodes of a clique.

3. A SMALL SIZE TOY EXAMPLE

In order to illustrate the results presented so far we work out a small size example in details.



EXAMPLE 3.1. Let us consider the graph $G = (V, E)$. Here $V = \{1, \dots, 8\}$. The adjacency matrix of G is depicted in Table 1. Figure 1 shows a possible geometric representation of G .

Using the graph G we constructed the edges auxiliary graph Γ . Table 2 displays the adjacency matrix of Γ . The rows and columns of this adjacency matrix are labeled with ordered pairs. In order to avoid an overly cluttered table we suppressed the braces when we recorded the ordered pairs. For instance instead of $\{1, 2\}$ we wrote simply 1, 2. Figure 2 depicts a possible geometric representation of the edge auxiliary graph Γ .

Next we well colored the nodes of Γ using the greedy sequential coloring procedure. The computations are recorded in Table 3. The rows of the table are labeled by the nodes of Γ . In other words the first column of the table holds the nodes of Γ . We colored node $\{1, 2\}$ by color 1. (Node $\{1, 2\}$ must receive a color and we may assume that it receives color 1 since this is only a matter of exchanging the colors among each other.) Then we checked if node $\{1, 3\}$ may receive color 1, that is, we checked if node $\{1, 3\}$ is adjacent to node $\{1, 2\}$ in the complement of Γ . Next we checked if node $\{1, 4\}$ may receive color 1, that is, if node $\{1, 4\}$ is adjacent to nodes $\{1, 2\}$ and $\{1, 3\}$ in the complement of Γ . The adjacencies were recorded by marking the color entries by a “]” symbol. We continued in this way until each node is colored. The last column of the table records the colors of the nodes. The color classes are the following

$$\begin{aligned} C_1 &= \{ \{1, 2\}, \{1, 3\}, \{2, 3\} \}, \\ C_2 &= \{ \{1, 4\}, \{3, 4\} \}, \\ C_3 &= \{ \{1, 5\} \}, \\ C_4 &= \{ \{2, 6\}, \{2, 7\}, \{3, 6\}, \{3, 7\}, \{6, 7\} \}, \\ C_5 &= \{ \{4, 8\} \}, \\ C_6 &= \{ \{5, 6\}, \{5, 8\}, \{6, 8\} \}, \\ C_7 &= \{ \{7, 8\} \}. \end{aligned}$$

Applying Lemma 2.2 to the fourth color class gives a 4-clique Δ in the graph G with nodes 2,3,6,7. By Lemma 2.1, the 4-clique Δ in G gives rise to a 6-clique Ω in the edge auxiliary graph Γ with nodes $\{2, 3\}, \{2, 6\}, \{2, 7\}, \{3, 6\}, \{3, 7\}, \{6, 7\}$. The greedy well coloring of the nodes of the edge auxiliary did not provide a color class with 6 elements. In spite of this we were able to locate a 4-clique in G , that is a 6-clique in Γ . This observation illustrates the significance of Lemma 2.2

As a last step we well color the nodes of the edge auxiliary graph Γ . Table 4 shows that the nodes of Γ can be well colored using 6 colors. Obviously, the nodes of a clique must receive pair-wise distinct colors at a well coloring of the nodes of a graph. This implies that the clique number is less than or equal to the chromatic number for each finite simple graph. Therefore, the clique number of the edge auxiliary graph Γ is at most 6, that is, the clique number of G is at most 4. On the other hand we have located a 4-clique in Γ . The moral of this observation is that a well coloring of the nodes of the edge auxiliary graph can be used to assess how far is the size of the spotted suboptimal clique from the optimal clique size.

4. NUMERICAL EXPERIMENTS

When the graph G has, say 1500 nodes, then the associated edge auxiliary graph Γ may have as many as 1 million nodes. Locating a clique in G requires a well coloring the nodes of a large graph Γ . One might wonder if this proposal is reasonable at all. In this section we set out to assess the practicality of this approach.

For numerical testing purposes we have selected three infinite families of graphs. All these graph are coming from coding theory. They are connected to the existence and construction of certain error detecting and error correcting codes. The so-called monotonic matrices are in intimate connection with codes over the alphabet $\{1, \dots, n\}$. The code words all have length three. The problem is to find a code whose inner distance is at least two. (See [6, 7, 8].) The deletion error detecting codes are consisting of binary code words of length n . These words are transmitted over a noisy channel.

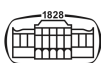


TABLE 3. Greedy sequential coloring of the nodes of the complement of the edge auxiliary graph Γ in Example 3.1.

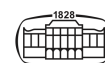
{1,2}	1	1]	1]	1	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]
{1,3}	←	1	1]	1	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]	1]
{1,4}		←	2]	2]	2]	2]	2]	2]	2]	2]	2]	2]	2]	2]	2]	2]	2]
{1,5}			←	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]
{2,3}				←	1	1	1]	1	1	1]	1]	1]	1]	1]	1]	1]	1]
{2,6}					←	4	4]	4]	4]	4]	4]	4]	4]	4]	4]	4]	4]
{2,7}						←	4	4]	4]	4]	4]	4]	4]	4]	4]	4]	4]
{3,4}							←	2]	2]	2]	2]	2]	2]	2]	2]	2]	2]
{3,6}								←	4	4]	4]	4]	4]	4]	4]	4]	4]
{3,7}									←	4]	4]	4]	4]	4]	4]	4]	4]
{4,8}										←	5]	5]	5]	5]	5]	5]	5]
{5,6}											←	6	6]	6]	6]	6]	6]
{5,8}												←	6]	6]	6]	6]	6]
{6,7}													←	4	4]	4]	4]
{6,8}														←	6	6]	6]
{7,8}															←	7	7]

TABLE 4. Greedy sequential coloring of the nodes of the edge auxiliary graph Γ in Example 3.1.

{1,2}	1]	1	1	1]	1	1	1	1	1	1	1	1	1	1	1	1	1
{1,3}	←	2]	2	2]	2	2	2]	2	2	2	2	2	2	2	2	2	2
{1,4}		←	1	1	1	1	1]	1	1	1	1	1	1	1	1	1	1
{1,5}			←	1	1	1	1	1	1	1	1	1	1	1	1	1	1
{2,3}				←	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]	3]
{2,6}					←	1]	1	1]	1]	1	1	1]	1]	1]	1]	1]	1]
{2,7}						←	2	2]	2]	2	2	2]	2]	2]	2]	2]	2]
{3,4}							←	3	3]	3]	3]	3]	3]	3]	3]	3]	3]
{3,6}								←	4]	4	4]	4]	4]	4]	4]	4]	5]
{3,7}									←	5	5]	5]	5]	5]	5]	5]	5]
{4,8}										←	1	1]	1]	1]	1]	1]	1]
{5,6}											←	1]	1]	1]	1]	1]	1]
{5,8}												←	2	2]	2]	2]	2]
{6,7}													←	6]	6]	6]	6]
{6,8}														←	3]	3]	3]
{7,8}															←	1	1]

Due to transmission error on the receiver side a shorter word may arrive. The task is to devise a code that makes possible to detect a one bit deletion error. (For further details see [5].) The Johnson codes that are considered here are binary codes with word length n . Each code word consists of 4 1's and $n - 4$ 0's. The Hamming distance of two distinct code words is at least 3.

The results of the numerical experiments are summarized in the Tables 5, 6, 7. We describe the meaning of the entries using the 13-th row of Table 5 as an illustration. A graph G is associated with a monotonic matrix of parameter $n = 13$. The graph has $|V| = 2\ 197$ vertices. The associated edge auxiliary graph Γ has $|E| = |W| = 2\ 135\ 434$ nodes. These numbers occupy the cells in the first three columns of the row. The clique we have spotted has 40 nodes and the last column contains this value. We used the simple sequential greedy coloring procedure to get a well coloring of the nodes of the edge auxiliary graph Γ .



Let us choose a node v of the graph G whose degree is minimum. If the degree of v is equal to $|V| - 1$, then the graph G itself is a clique. If the degree of v is less than $|V| - 1$, then we delete v from the graph G . Repeating this procedure in connection with this new graph finally we end up with a clique Δ in G . The size of Δ is a lower bound for the clique number $\omega(G)$ of G . We may refer to this clique locating algorithm as the minimum degree rule. The lower estimate provided by the minimum degree rule is in the column headed by "min" in Table 5.

TABLE 5. Monotonic matrices.

n	$ V $	$ E $	est	min	max
3	27	189	5	4	4
4	64	1 296	8	6	8
5	125	5 500	9	9	10
6	216	17 550	12	13	13
7	343	46 305	17	15	15
8	512	106 624	20	19	21
9	729	221 616	23	22	25
10	1 000	425 250	27	25	31
11	1 331	765 325	32	29	33
12	1 728	1 306 800	35	31	38
13	2 197	2 135 484	40	36	43
14	2 744	3 362 086	44	37	49
15	3 375	5 126 665	47	46	56
16	4 096	7 603 200	54	47	59

Let us pick a node v of the graph G with maximum degree. Then let us restrict G to the set of neighbors of v in G . Repeating this procedure in connection with this new graph eventually we end up with a clique Δ in G . The size of Δ is a lower estimate of the clique number $\omega(G)$ of G . We may refer to this this clique spotting algorithm as the maximum degree rule. In Table 5 the column labeled by "max" contains the lower estimate of $\omega(G)$ established by the maximum degree rule.

We may conclude that the algorithm seems to work in connection with non-trivial size graphs in a reliable manner. However, there are many algorithms to locate suboptimal cliques in a given graph. Some of these procedures may provide better estimates. Only after working with the algorithm for a longer period of time involving a wider variety and range of graphs would provide more information about the merits of the proposed procedure.

TABLE 6. Deletion error correcting codes.

n	$ V $	$ E $	est	min	max
3	8	9	2	2	2
4	16	57	4	3	4
5	32	305	6	5	6
6	64	1 473	9	8	10
7	128	6 657	15	14	15
8	256	28 801	25	22	25
9	512	121 089	42	38	43
10	1 024	499 713	72	65	74
11	2 048	2 037 761	127	113	131
12	4 096	8 247 297	229	197	234

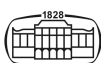


TABLE 7. Johnson codes.

n	$ V $	$ E $	est	min	max
6	15	45	3	2	3
7	35	385	7	6	7
8	70	1 855	14	8	14
9	126	6 615	16	15	14
10	210	19 425	24	20	30
11	330	49 665	32	29	30
12	495	114 345	45	39	50
13	715	242 385	57	47	58
14	1 001	480 480	77	65	76
15	1 361	900 900	105	76	93
16	1 820	1 611 610	140	97	126
17	2 380	2 769 130	140	113	144
18	3 060	4 594 590	169	138	183
19	3 876	7 393 470	201	168	210

REFERENCES

- [1] ÇATALZÜREK, Ü. V., FEO, J., GEBREMEDHIN, A. H., HALAPPANAVAR, M., AND POTHEN, A. Graph coloring algorithms for multi-core and massively multi threaded architectures. *Parallel Comp.* 38 (2012), 576–594.
- [2] LEIGHTON, F. T. A graph coloring algorithm for large scheduling problems. *Journal of Research of National Bureau of Standards* 84 (1979), 489–506.
- [3] MATULA, D. W., MARBLE, G., AND ISAACSON, J. Graph coloring algorithms. In *Graph Theory and Computing*, R. Read, Ed. Academic Press, 1972, pp. 109–122.
- [4] PAPADIMITRIOU, C. H. *Computational Complexity*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1994.
- [5] SLOANE, N. J. A. Challenge problems: Independent sets in graphs. <http://neilsloane.com/doc/graphs.html>.
- [6] SZABÓ, S. Parallel algorithms for finding cliques in a graph. *Journal of Physics, Conference Series* 268 (2011), 012030. doi:10.1088/1742-6596/268/1/012030.
- [7] SZABÓ, S. Monoton matrices and finding cliques in a graph. *Annales Univ. Sci. Budapest., Sect. Computatorica* 41 (2013), 307–322.
- [8] WEISSTEIN, E. W. Monotonic matrix. in MathWorld–A Wolfram Web Resource.

Open Access statement. This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (<https://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited, a link to the CC License is provided, and changes – if any – are indicated. (SID_1)

