Serdica

Mathematical Journal

Сердика

Математическо списание

Serdica Math. J. **47** (2021), 31–44

Serdica
Mathematical Journal

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

CLIQUE NUMBER ESTIMATE BASED ON COLORING OF THE NODES

Sándor Szabó, Bogdán Zavalnij

Communicated by V. Drensky

Abstract. We will describe an algorithm to establish an upper estimate of the clique number of a given graph. The procedure is based on greedy legal coloring of the nodes. In order to assess the performance of the procedure we carried out a large scale numerical experiment.

1. Introduction. Let G be a finite simple graph. The adjective "finite" refers to the fact that the graph G has finitely many nodes and finitely many edges. The simplicity means that G does not have any loop or double edges. A finite simple graph G can be uniquely described by the ordered pair (V, E), where V is the set of vertices of G 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. If the clique Δ has k nodes we will say that the size of Δ is k. Shortly we also will say that Δ is a k-clique in G. A clique Δ is maximal

2020 *Mathematics Subject Classification:* Primary 05C15; Secondary 05B45, 52C22.

Key words: Clique number, chromatic number, maximum clique, greedy coloring, clique number estimates.

if it cannot be extended to a larger clique in G by adding a node of G to Δ. A k-clique Δ in G is a maximum clique if G does not contain any $(k+1)$-clique. A graph G may contain maximal cliques of various sizes. But all the maximum cliques of G have the same size. This well defined number is called the clique number of G and it is denoted by $\omega(G)$.

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

The reader will notice that double edges do not influence the coloring while a loop obstructs any legal coloring. So the decision to restrict our attention to simple graphs is a sensible one. For a finite simple graph G there is a well defined positive integer k such that the nodes of G can be colored legally using k colors and the nodes of G cannot be legally colored using $k-1$ colors. This k is called the chromatic number of G and it is denoted by $\chi(G)$.

The complexity theory of computations is teaching us that computing $\omega(G)$ is computationally challenging. Similarly, determining $\chi(G)$ is a hard computational problem. (See [6].) However, there are practically important problems that depend on computing $\omega(G)$ or $\chi(G)$. In certain situations locating a large but not necessarily optimal clique in G instead of finding $\omega(G)$ is satisfactory. Similarly, legally coloring the nodes of G using not necessarily the optimal number of colors has practical utility. It was pointed out in [9] that algorithms locating cliques in a graph can be used to estimate the chromatic number of a graph. In this note we explore the other direction that drawing on the available greedy coloring procedures we can estimate the clique number of a graph.

The nodes of a clique must receive pair-wise distinct colors at a legal 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. There are graphs for which the chromatic number is a poor estimate of the clique number. Namely, for each positive integer k there is a 3-clique free graph whose chromatic number is equal to k. (See [4].) The number of the vertices of these graphs increases exponentially in terms of k. So only for very large graph we may expect large gap between the clique and chromatic numbers.

Computing the chromatic number of a given graph is an NP-hard problem. This is why we use legal colorings with not necessarily optimal number of colors

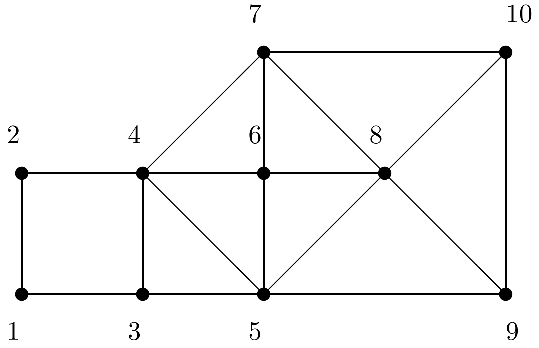

Fig. 1. A geometric representation of the graph G in Example 1.

Table 1. The adjacency matrix of the graph G in Example 1.

	1	2	3	4	5	6	7	8	9	10
1	×	●	●							
2	●	×		●						
3	●		×	●	●					
4		●	●	×	●	●	●			
5			●	●	×	●		●	●	
6				●	●	×	●	●		
7				●		●	×	●		●
8					●	●	●	×	●	●
9					●			●	×	●
10							●	●	●	×

provided by a greedy coloring procedure instead of determining the chromatic number. This opens up a further source of inaccuracy in the clique number estimate. Not only the chromatic number is not a tight estimate of the clique number but in addition we use a not very accurate estimate of the chromatic number itself. The simplest greedy node coloring algorithm maybe well off the mark. The reason is that there are graphs with $2n$ vertices whose the chromatic number is 2 and in certain settings (using a specific ordering of the nodes of the graph) the simplest greedy coloring algorithm can end up with using n colors.

2. The basic estimate. Let $G = (V, E)$ be a finite simple graph such that $|V| = n$ and $v_1, \dots, v_n$ are all the nodes of G. Let H_i be the subgraph of G induced by the set $N(v_i)$. Here $N(v)$ stands for the set of neighbors of the node v. In notation $N(v) = \{u : \ u \in V, \ \{u, v\} \in E\}$. In order the present the estimating procedure we choose an upper bound $\alpha(i)$ of the clique number $\omega(H_i)$

of H_i. In other words we choose an $\alpha(i)$ for which the inequality $\omega(H_i) \leq \alpha(i)$ holds. At this moment the reader need not to worry how good is this estimate and how one can arrive at it. We assume that the vertices of G are arranged in such a way that the conditions $\alpha(1) \geq \cdots \geq \alpha(n)$ are satisfied. This situation can be reached by first picking the upper bounds of the clique sizes of the subgraphs then rearrange the subgraphs such that the corresponding upper bounds are in decreasing order.

Let K_i be the subgraph of G induced by the set of nodes $\{v_1, \ldots, v_i\}$. The function that assigns $\omega(K_i)$ to i we will call the Östergård function of the graph G with respect to a fixed ordering of the nodes of G. This function is introduced by P. R. J. Östergård in connection with the maximum clique problem in [5]. Let $\beta(i)$ be an upper bound of $\omega(K_i)$. In other words $\beta(i)$ is a number for which $\omega(K_i) \leq \beta(i)$. We assume that $\beta(1) \leq \cdots \leq \beta(n)$, that is the function β is an increasing function in the range $1, \ldots, n$. If the β function we pick by accident were not increasing, say $\beta(i-1) > \beta(i)$ would hold, then we may replace $\beta(i)$ by $\beta(i-1)$. The condition $\omega(K_i) \leq \beta(i)$ is not violated. Finally we set $\gamma(i) = \min\{\alpha(i)+1, \beta(i)\}$ for each i, $1 \leq i \leq n$.

Lemma 1. *With the notations we introduced $\omega(G) \leq \max\{\gamma(1), \ldots, \gamma(n)\}$ holds.*

Proof. Note that the graph K_1 has only one vertex v_1 and so $\omega(K_1) = 1$. The graph K_n is identical with G and so $\omega(K_n) = \omega(G)$. The Östergård function is an increasing function on the range $1, \ldots, n$, that is, $\omega(K_1) \leq \cdots \leq \omega(K_n)$. Plainly, $\omega(G) = \max\{\omega(K_1), \ldots, \omega(K_n)\}$.

If $\beta(i) \leq \alpha(i) + 1$ for each i, $1 \leq i \leq n$, then $\gamma(i) = \beta(i)$ for each i, $1 \leq i \leq n$. It follows that $\omega(K_i) \leq \beta(i) = \gamma(i)$ for each i, $1 \leq i \leq n$. In particular $\omega(K_n) \leq \gamma(n)$ and so $\omega(G) = \omega(K_n) \leq \max\{\gamma(1), \ldots, \gamma(n)\}$. For the remaining part of the proof we may assume that there is an r such that $1 \leq r \leq n$, $\beta(r) \leq \alpha(r)+1$ and $\beta(i) > \alpha(i)+1$ for each i, $r+1 \leq i \leq n$. Thus $\gamma(i) = \alpha(i)+1$ for each i, $r+1 \leq i \leq n$.

The function α decreases in the range $1, \ldots, n$. The function γ increases in the range $1, \ldots, r$ and it decreases in the range $r+1, \ldots, n$. Thus

$$\begin{aligned} \max\{\gamma(1), \ldots, \gamma(n)\} &= \max\{\gamma(r), \gamma(r+1)\} \\ &= \max\{\beta(r), \alpha(r+1)+1\}. \end{aligned}$$

If $\omega(K_i) \leq \alpha(i)+1$ for each i, $r+1 \leq i \leq n$, then

$$\begin{aligned}\omega(G) &\leq \max\{\gamma(r), \gamma(r+1)\} \\ &= \max\{\gamma(1), \dots, \gamma(n)\}.\end{aligned}$$

For the remaining part of the proof we may assume that there is an s such that $r+1 \leq s \leq n$, $\omega(K_s) \leq \alpha(s)+1$ and $\omega(K_i) > \alpha(i)+1$ for each i, $s+1 \leq i \leq n$. We will show that the Östergård function is constant in the range $s+1, \dots, n$.

Let L_i be the subgraph of G induced by the set of nodes $N(v_i) \cap \{v_1, \dots, v_i\}$. As L_i is a subgraph of H_i we get $\omega(L_i) \leq \omega(H_i)$. The Östergård function can jump only one in one step. Note that

$$\omega(K_i) = \begin{cases} \omega(K_{i-1}), & \text{only if } \omega(L_i) < \omega(K_{i-1}), \\ \omega(K_{i-1})+1, & \text{only if } \omega(L_i) = \omega(K_{i-1}). \end{cases}$$

Let us assume on the contrary that the Östergård function is jumping. The computation

$$\begin{aligned}\omega(K_i) &= \omega(K_{i-1})+1 \\ &= \omega(L_i)+1 \\ &\leq \omega(H_i)+1 \\ &\leq \alpha(i)+1\end{aligned}$$

shows that $\omega(K_i) \leq \alpha(i)+1$. But this contradicts to our assumption that $\omega(K_i) > \alpha(i)+1$. Therefore in the range $s+1, \dots, n$ the Östergård function stays constant. It follows that $\omega(K_i) \leq \alpha(s)+1$ for each i, $s \leq i \leq n$. In particular $\omega(G) = \omega(K_n) \leq \alpha(s)+1$. Using $\alpha(s)+1 \leq \beta(s)$ we get $\omega(G) \leq \gamma(s)$. Thus

$$\begin{aligned}\omega(G) &\leq \gamma(s) \\ &= \max\{\gamma(r), \gamma(r+1)\} \\ &= \max\{\gamma(1), \dots, \gamma(n)\}.\end{aligned}$$ □

The graphs K_i, L_i are subgraphs of the given graph G. Lemma 1 can be viewed such that the clique number estimates of the graphs K_i, L_i can be combined together to get a clique number estimate for the larger graph G. From local informations we can piece together global information.

Table 2. Nodes with their neighbors

node	neighbors	color
1	2,3	1
2	1,4	1
3	1,4,5	2
4	2,3,5,6,7	2
5	3,4,6,8,9	2
6	4,5,7,8	2
7	4,6,8,10	2
8	5,6,7,9,10	3
9	5,8,10	2
10	7,8,9	2

Table 3. The $\alpha(i)+1$ and $\beta(i)$ values

i	v_i	$\alpha(i)$	$\alpha(i)+1$	nodes of L_i	colors	$\beta(i)$
1	8	3	4		0	1
2	3	2	3		0	1
3	4	2	3	3	1	2
4	5	2	3	3,4,8	2	3
5	6	2	3	4,5,8	2	3
6	7	2	3	4,6,8	2	3
7	9	2	3	5,8	2	3
8	10	2	3	7,8,9	2	3
9	1	1	2	3	1	3
10	2	1	2	1,4	1	3

The clique number of the graphs can be estimated in many ways. In addition the computations can be done independently of each other. This opens up a way to construct parallel algorithms. Here we make some comments about the computations we have experimented with. Let $\delta(i)$ be an upper estimate of the clique size of the graph L_i. A possible candidate for the upper estimate $\alpha(i)$ can be computed by legally coloring the nodes of K_i. Similarly, a possible $\delta(i)$ can be computed by legally coloring the nodes of L_i. We used $\delta(i)$ to construct the numbers $\beta(i)$. We set $\beta(1)=1$. Then for each i, $2 \leq i \leq n$ we set

$$\beta(i) = \begin{cases} \beta(i-1), & \text{if } \delta(i) < \beta(i-1), \\ \beta(i-1)+1, & \text{if } \delta(i) \geq \beta(i-1). \end{cases}$$

There is a large variety of well implemented greedy coloring algorithms.

The majority of them are sequential but there are parallel algorithms as well. (See [3], [1].) The most trivial vertex coloring algorithm is the algorithm that assigns pair-wise distinct colors to the nodes of the graph. Needless to say that one cannot expect good clique size estimates from this approach. In our numerical experiments we used the so-called RLF coloring scheme. (See [2].)

3. A small size toy example. In order to illustrate the points of the argument presented in Section 2 we work out a small example in details.

Example 1. Let us consider the graph $G = (V, E)$. Here $V = \{1, \ldots, 10\}$. The adjacency matrix of G is depicted in Table 1. Figure 1 shows a geometric representation of G.

The chromatic number of the graph G is equal to 4. The nodes of G can be legally colored using 4 colors. One possible coloring is the following.

node	1	2	3	4	5	6	7	8	9	10
color	1	2	2	1	3	2	3	1	2	4

On the other hand the nodes 5, 6, 7, 10, 9 are situated on a cycle of odd length and so they cannot be colored legally with less than three colors. The node 8 is a common neighbor of these vertices. Therefore these six vertices must receive at least four colors at a legal coloring. The graph G has several 3-cliques. This means that the clique number of G is at least 3 The procedure for estimating the clique size will reveal that $\omega(G) \leq 3$ and so $\omega(G) = 3$.

The row labeled by 4 in Table 2 contains the information that the neighbors of the node 4 of the graph G are the nodes 2, 3, 5, 6, 7. The nodes of the subgraph of G induced by these nodes can be legally colored using 2 colors. The node 4 is in the 1-st column. Its neighbors are in the 2-nd column. The number of colors used in a legal coloring of the nodes stands in the 3-rd column. An inspection of Figure 1 reveals that the subgraph induced by the set of neighbors of the node 4 consists of a path and two isolated points. The nodes obviously can be legally colored using two colors. The other nodes of the graph G can be handled in a similar way.

The 2-nd column of Table 3 contains the vertices of the graph G. This time the nodes are listed in such order that the α function is decreasing in the range $1, \ldots, 10$. For each node v_i we computed $N(v_i) \cap \{v_1, \ldots, v_i\}$ and listed the elements of this set in the 5-th column. After legally coloring the nodes of the corresponding graph L_i we filled the 6-th column with the $\delta(i)$ values. Finally, we computed the $\beta(i)$ number which are recorded in the last column of the table.

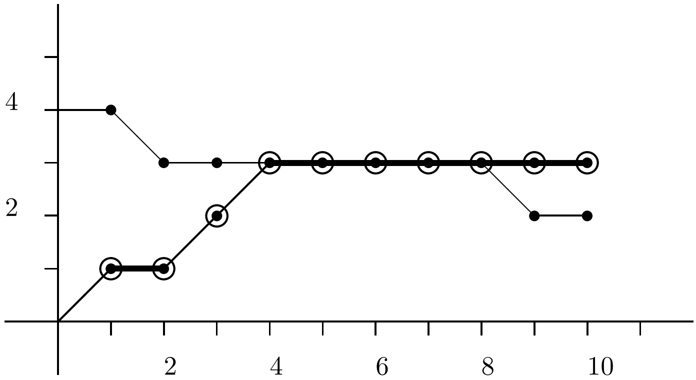

Fig. 2. A plot of the $\alpha(i) + 1$ and $\beta(i)$ values in Example 1

Figure 2 depicts the points $(i, \alpha(i)+1)$, $(i, \beta(i))$ in a Cartesian coordinate system. For a better visual effect we connected the points by straight line sections. The α, β functions are defined at discrete points of the number scale. Here we extended these functions to be continuous (piece-wise linear) functions.

From the example we can see that in the course of the computation we may meet situation when some of the graphs H_i, K_i, L_i does not have any nodes. The clique and chromatic numbers of the empty graph by definition are equal to 0.

4. Numerical experiments. We have selected three infinite families of graphs. The graphs are related to the existence and construction of certain error detecting and error correcting codes. Our choices were motivated by the facts that the clique number of a few of these graphs are known. Estimating the unknown clique numbers is contributing to the knowledge on this field. The monotonic matrices are connected to codes over the alphabet $\{1, \ldots, n\}$. Each code words has length three. The problem is to find a code whose inner distance is at least two. (See [8], [10].) The deletion error detecting codes are consisting of binary code words of length n. These words are sent over a noisy channel. Due to transmission error on the receiver side a shorter word may arrive. The task is devise a code that makes possible to detect a one bit deletion error. (For further details see [7].) The Johnson codes we are considering 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.

In the course of our numerical experiments we made a number of observations. Some of these suggested possible improvement on the original approach.

Let us consider a node v_i of the graph G and let $\alpha(i)$ be the upper estimate we computed for $\omega(H_i)$. If v_i has less than $\alpha(i)$ neighbors whose α value is at least $\alpha(i)$, then v_i cannot be a node of a clique of size $\alpha(i)$ in G. Consequently, the value $\alpha(i)$ associated with the node v_i can be reduced. In short, we may replace $\alpha(i)$ by $\alpha(i) - 1$.

Based on this observation after computing the $\alpha(i)$ values we incorporated an adjustment phase in which we inspected the neighbors of each node to spot if the $\alpha(i)$ value could be reduced. One modification in an $\alpha(i)$ could trigger further reduction so the reduction phase is repeated until a stationary situation is reached.

The next observation is related to the fact that the greedy coloring algorithm we used is sensitive to the order the nodes of the graph are listed. The graph L_i is a subgraph of H_i. Consequently $\omega(L_i) \leq \omega(H_i)$ must hold. However, when we estimate $\omega(L_i)$, $\omega(H_i)$ by the number of colors in a legal coloring it may happen that we get a worst upper bound for $\omega(L_i)$ than for $\omega(H_i)$. The ordering of the nodes of the two graph maybe different and this is the reason of this behavior. We have incorporated an inspection into our algorithm. If by accident $\delta(i) > \alpha(i)$, then we replaced $\delta(i)$ by $\alpha(i)$.

The third observation we used is a bit less obvious. From the given graph G we constructed a larger auxiliary graph Γ. We estimated the clique number of the auxiliary graph Γ to get an estimate for the clique number of G.

Let us fix a positive integer l. Using the graph $G = (V, E)$ and the integer l we construct a new graph $\Gamma = (W, F)$. We will call Γ the l-fold auxiliary graph of G. The nodes of Γ are the ordered pairs (v, r), where $v \in V$ and $1 \leq r \leq l$. Clearly Γ has $|V|l$ nodes. Whenever the unordered pair $\{u, v\}$ is an edge of G we add the unordered pair $\{(u, r), (v, s)\}$ as an edge to Γ for each r, s, $1 \leq r, s \leq l$. Further we add the unordered pair $\{(v, r), (v, s)\}$ as an edge to Γ for each $v \in V$ and for each r, s, $1 \leq r < s \leq l$.

For the sake of an easier reference we state the connection between the the graphs G and Γ we will use in the form of a lemma.

Lemma 2. *If G has a clique of size k, then Γ has a clique of size kl.*

Proof. Let Δ be a clique in G such that $u_1 \ldots, u_k$ are all the nodes of Δ. We claim that the nodes

$$(u_1, 1), \ldots, (u_k, 1), \ldots, (u_1, l), \ldots, (u_k, l)$$

are the nodes of a clique in Γ. The unordered pair $\{u_i, u_j\}$ is an edge of G for

Table 4. Monotonic matrices

n	$\lvert V\rvert$	$\lvert E\rvert$	1-fold	2-fold	3-fold	4-fold
3	27	189	5	5	5	5
			5	10	15	20
4	64	1 296	9	8	8	8
			9	17	25	34
5	125	5 500	14	13	13	12
			14	26	39	51
6	216	17 550	21	20	19	19
			21	40	57	76
7	343	46 305	30	27	26	26
			30	55	80	104
8	512	106 624	40	36	35	34
			40	73	105	138
9	729	221 616	52	47	45	44
			52	94	135	176
10	1 000	425 250	65	59	56	55
			65	118	170	220
11	1 331	765 325	80	72	69	67
			80	145	207	270
12	1 728	1 306 800	96	87	83	81
			96	175	249	324
13	2 197	2 135 484	114	103	98	96
			114	207	295	384
14	2 744	3 362 086	133	121	115	112
			133	243	346	449
15	3 375	5 126 665	155	140	133	130
			155	281	400	520
16	4 096	7 603 200	177	161	153	148
			177	323	459	595
17	4 913	11 005 120	201	183	174	169
			201	367	522	676
18	5 832	15 590 394	227	207	196	190
			227	414	589	763
19	6 859	21 667 581	256	232	220	219
			256	465	660	855

Table 5. Deletion error correcting codes

n	$\|V\|$	$\|E\|$	1-fold	2-fold	3-fold	4-fold
3	8	9	2	2	2	2
			2	4	6	8
4	16	57	4	4	4	4
			4	8	12	16
5	32	305	6	6	6	6
			6	12	18	24
6	64	1 473	11	10	10	10
			11	21	32	42
7	128	6 657	21	19	19	18
			21	38	57	75
8	256	28 801	41	36	34	33
			41	72	103	135
9	512	121 089	79	67	63	61
			79	135	190	247
10	1 024	499 713	153	128	118	114
			153	257	355	456
11	2 048	2 037 761	293	243	222	211
			293	487	667	846
12	4 096	8 247 297	563	463	420	396
			563	926	1 260	1 585
13	8 192	33 222 657	1 077	881	796	747
			1 077	1 763	2 388	2 991

each i, j, $1 \leq i < j \leq k$. Consequently, the unordered pair $\{(u_i, r), (u_j, s)\}$ is an edge of Γ for each i, j, $1 \leq i < j \leq k$ and for each r, s, $1 \leq r, s \leq l$. In the $i = j$ case the unordered pair $\{(u_i, r), (u_i, s)\}$ is an edge of Γ for each i, $1 \leq i \leq k$ and for each r, s, $1 \leq r < s \leq l$. □

The content of Lemma 2 is that if we establish an upper bound for $\omega(\Gamma)$ for a given value of l, then dividing this upper bound by l provides an upper bound for $\omega(G)$. Estimating $\omega(\Gamma)$ is computationally more expensive than estimating $\omega(G)$ directly. (Using the same program with the inputs Γ and G.) But the more expensive estimate is typically sharper.

The results of the numerical experiments are summarized in the Tables 4, 5, 6. We describe the meaning of the entries in the last row of Table 4. A graph G is associated with a monotonic matrix of parameter $n = 17$. The graph has $|V| = 4\ 913$ vertices and $|E| = 11\ 005\ 120$ edges. These numbers occupy the cells in the first three columns of the row. We fix a positive integer l such that

Table 6. Johnson codes

n	$\lvert V\rvert$	$\lvert E\rvert$	1-fold	2-fold	3-fold	4-fold
6	15	45	3 3	3 6	3 9	3 12
7	35	385	7 7	7 14	7 21	7 29
8	70	1 855	16 16	14 28	14 42	14 56
9	126	6 615	25 25	24 48	22 66	21 87
10	210	19 425	42 42	37 74	33 101	30 121
11	330	49 665	63 63	53 107	48 144	41 165
12	495	114 345	89 89	74 149	65 195	55 220
13	715	242 385	121 121	99 199	86 258	71 286
14	1 001	480 480	160 160	129 258	111 333	91 364
15	1 361	900 900	206 206	164 328	140 420	113 455
16	1 820	1 611 610	259 259	204 408	173 521	140 560
17	2 380	2 769 130	322 322	251 502	212 636	170 680
18	3 060	4 594 590	392 392	304 609	256 768	203 816
19	3 876	7 393 470	473 473	365 730	305 916	242 969
20	4 895	11 579 550	565 565	433 866	360 1 082	285 1 140
21	5 985	17 722 105	665 665	523 1 046	425 1 275	358 1 436

$1 \leq l \leq 4$ and construct an l-fold auxiliary graph Γ from G. The algorithm gives that $\omega(\Gamma) \leq 676$ for $l = 4$. This information can be found in the last cell in the row. By Lemma 2, it follows that $\omega(G) \leq 676/4 = 169$. Other entries in the table can be interpreted in an analogous way.

Only working with the algorithm for a longer period of time involving a much wider variety and range of graphs would provide a base to assess the merits of the proposed procedure. At this stage we may conclude that the algorithm works in connection with non-trivial size graphs in a reliable manner and provides far from trivial upper bounds for the clique numbers.

REFERENCES

[1] Ü. V. Çatalyürek, J. Feo, A. H. Gebremedhin, M. Halappanavar, A. Pothen. Graph coloring algorithms for multi-core and massively multithreaded architectures. *Parallel Comput.* **38**, sl 10–11 (2012), 576–594.

[2] F. T. Leighton. A graph coloring algorithm for large scheduling problems. *J. Res. Nat. Bur. Standards* **84**, *6* (1979), 489–506.

[3] D. W. Matula, G. Marble, J. Isaacson. Graph coloring algorithms. In: Graph Theory and Computing (ed. R. Read) New York, Academic Press, 1972, 109–122.

[4] J. Mycielski. Sur le coloriage des graphes. *Colloq. Math.* **3** (1955), 161–162 (in French).

[5] P. R. J. Östergård. A fast algorithm for the maximum clique problem. Sixth Twente Workshop on Graphs and Combinatorial Optimization (Enschede, 1999). *Discrete Appl. Math.* **120**, *1–3* (2002), 197–207.

[6] C. H. Papadimitriou. Computational Complexity. Reading, MA, Addison-Wesley Publishing Company, Inc., 1994.

[7] N. J. A. Sloane. Challenge Problems: Independent Sets in Graphs. `https://oeis.org/a265032.html`

[8] S. Szabó, Monotonic matrices and clique search in graphs, *Ann. Univ. Sci. Budapest., Sect. Comput.* **41** (2013), 307–322.

[9] S. Szabó and B. Zaválnij, Reducing graph coloring to clique search, *Asia Pacific Journal of Mathematics* **3** (2016), 64–85, `http://apjm.apacific.org/PDFs/3-1-64-85.pdf`.

[10] E. W. Weisstein. Monotonic Matrix. In: MathWorld–A Wolfram Web Resource. `http://mathworld.wolfram.com/MonotonicMatrix.html`

Sándor Szabó
Institute of Mathematics and Informatics
University of Pécs
Ifjúság u. 6
7624 Pécs, Hungary
e-mail: `sszabo7@hotmail.com`

Bogdán Zavalnij
Alfréd Rényi Institute of Mathematics
Budapest, Hungary
e-mail: `bogdan@renyi.hu`

Received January 11, 2021