Acta Univ. Sapientiae Informatica **13,** 1 (2021) 122–133

DOI: 10.2478/ausi-2021-0006

Estimating the fractional chromatic number of a graph

Sándor SZABÓ
University of Pécs
email: sszabo7@hotmail.com

Abstract. The fractional chromatic number of a graph is defined as the optimum of a rather unwieldy linear program. (Setting up the program requires generating all independent sets of the given graph.) Using combinatorial arguments we construct a more manageable linear program whose optimum value provides an upper estimate for the fractional chromatic number. In order to assess the feasibility of the proposal and in order to check the accuracy of the estimates we carry out numerical experiments.

1 Introduction and preliminaries

A loop and double edge free graph with finitely many vertices and edges is referred as a finite simple graph. Let V and E be the set of vertices and edges of a finite simple graph G, respectively. Clearly, the ordered pair (V, E) determines G uniquely.

Let $\mathsf{G} = (\mathsf{V}, \mathsf{E})$ be a finite simple graph. A subset I of V is called an independent set if any two distinct vertices of I are non-adjacent in G. We say that a subset C of V is a clique in G if two distinct vertices of C are always adjacent in G. If C has k elements we call C a k-clique. We would like to point out that a one element subset of V is both a clique and an independent set

Computing Classification System 1998: G.2.2
Mathematics Subject Classification 2010: 11A05, 11R04
Key words and phrases: clique and chromatic numbers, fractional chromatic number, integer programming.

of G. Similarly, the empty set is both a clique and an independent set of G. Sometimes the subgraph induced by C in G is called a clique. This ambiguity in the terminology is not going to cause any problem.

For each finite simple graph G there is an integer k such that G contains a k-clique but G does not contain any $(k+1)$-clique. This well defined k is called the clique number of G and it is denoted by $\omega(G)$. We color the nodes of G choosing colors from a palette of k colors such that each vertex of G receives exactly one color and the end points of adjacent vertices never receive the same color. Such a coloring of the nodes of G is called a legal k-coloring. For each finite simple graph G there is an integer k such that the nodes of G can be legally colored with k colors but they cannot be legally colored using $(k-1)$ colors. This well defined k is called the chromatic number of G and it is denoted by $\chi(G)$.

There are important problems in applied and theoretical discrete optimization which are essentially about determining the clique or chromatic numbers of a given graph. (Many such instances can be found in [1].) It is known from the complexity theory of the computations that the optimization problems of computing the clique or chromatic numbers belong to the NP hard complexity class. (For more details see [2], [4].) A commonly held interpretation of this fact is that computing the clique or chromatic numbers are computationally demanding tasks.

Let $G = (V, E)$ be a finite simple graph. If a vertex v of G is adjacent to each vertex of G distinct from v, then we call v a full degree node. Deleting a full degree node from G is reducing both the clique and chromatic number of G by one. From this reason we may restrict our attention to graphs without full degree nodes when we are looking for the clique or chromatic numbers.

Let $G = (V, E)$ be a full degree free finite simple graph and let $I(1), \dots, I(r)$ be all the independent sets of G. To each vertex v of G we assign a non-negative variable $x(v)$. The optimum value of the linear program P

$$\sum_{v \in V} x(v) \to \max$$

$$\sum_{v \in I(s)} x(v) \le 1, \quad 1 \le s \le r$$

is called the fractional clique number of G and it is denoted by $\omega_f(G)$. The optimum value of the dual of the program is referred to as fractional chromatic number of the graph G and $\chi_f(G)$ denotes it. By the duality theorem of linear programming, $\omega_f(G) = \chi_f(G)$.

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

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

Table 1: The adjacency matrices of the graph G and its complement $\overline{G}$ in Example 2.

Note that the linear program P is a relaxed version of the linear program Q

$$\sum_{v \in V} x(v) \to \max$$

$$x(u) + x(v) \leq 1, \quad \{u, v\} \notin E.$$

Solving Q in zero-one variables gives that $\omega(G) \leq \omega_f(G)$. The main reason behind restricting our attention to full degree free graphs is that in this situation the inequality $x(v) \leq 1$ holds for each $v \in V$.

Consider a legal coloring of the vertices of G and let $C(1), \ldots, C(k)$ be the color classes of the nodes. Note that the linear program R

$$\sum_{v \in V} x(v) \to \max$$

$$\sum_{v \in C(s)} x(v) \leq 1, \quad 1 \leq s \leq k$$

is a relaxed version of the linear program P. The optimum value of the dual of R is k even if we solve it in zero-one variables. It follows that $\chi_f(G) \leq \chi(G)$. Therefore

$$\omega(G) \leq \omega_f(G) = \chi_f(G) \leq \chi(G).$$

	16	17	18	24	25	28	35	38	45	46	47	57
1,6	×											
1,7		×										
1,8			×									
2,4				×	•				•			
2,5				•	×				•			
2,8						×						
3,5							×					
3,8								×				
4,5				•	•				×		•	•
4,6										×		
4,7									•		×	•
5,7									•		•	×

	16	17	18	24	25	28	35	38	45	46	47	57
1,6	×	•	•	•	•	•	•	•	•	•	•	•
1,7	•	×	•	•	•	•	•	•	•	•	•	•
1,8	•	•	×	•	•	•	•	•	•	•	•	•
2,4	•	•	•	×		•	•	•		•	•	•
2,5	•	•	•		×	•	•	•		•	•	•
2,8	•	•	•	•	•	×	•	•	•	•	•	•
3,5	•	•	•	•	•	•	×	•	•	•	•	•
3,8	•	•	•	•	•	•	•	×	•	•	•	•
4,5	•	•	•			•	•	•	×	•		
4,6	•	•	•	•	•	•	•	•	•	×	•	•
4,7	•	•	•	•	•	•	•	•		•	×	
5,7	•	•	•	•	•	•	•	•		•		×

Table 2: The adjacency matrices of the edge auxiliary graph Γ of the graph $\overline{G}$ and its complement $\overline{\Gamma}$ in Example 2.

{1,6}	[1	[1	[1	[1	[1	[1	[1	[1	[1	[1	[1	1
{1,7}	←	[2	[2	[2	[2	[2	[2	[2	[2	[2	[2	2
{1,8}		←	[3	[3	[3	[3	[3	[3	[3	[3	[3	3
{2,4}			←	4	[4	[4	[4	4	[4	[4	[4	4
{2,5}				←	[4	[4	[4	4	[4	[4	[4	4
{2,8}					←	[5	[5	[5	[5	[5	[5	5
{3,5}						←	[6	[6	[6	[6	[6	6
{3,8}							←	[7	[7	[7	[7	7
{4,5}								←	[4	4	4	4
{4,6}									←	[8	[8	8
{4,7}										←	9	9
{5,7}											←	9

Table 3: Greedy sequential coloring of the nodes of the complement of the edge auxiliary graph $\overline{\Gamma}$ in Example 2.

$C(1)$:	{1,6}	$I(1)$:	1,6
$C(2)$:	{1,7}	$I(2)$:	1,7
$C(3)$:	{1,8}	$I(3)$:	1,8
$C(4)$:	{2,4},{2,5},{4,5}	$I(4)$:	2,4,5
$C(5)$:	{2,8}	$I(5)$:	2,8
$C(6)$:	{3,5}	$I(6)$:	3,5
$C(7)$:	{3,8}	$I(7)$:	3,8
$C(8)$:	{4,6}	$I(8)$:	4,6
$C(9)$:	{4,7},{5,7}	$I(9)$:	4,5,7

Table 4: The cliques $C(1), \ldots, C(9)$ of the graph Γ and the corresponding cliques $I(1), \ldots, I(9)$ of the graph $\overline{G}$ in Example 2.

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8		
	1	1	1	1	1	1	1	1	$\rightarrow$	max
(1)	1					1			$\leq$	1
(2)	1						1		$\leq$	1
(3)	1							1	$\leq$	1
(4)		1		1	1				$\leq$	1
(5)		1						1	$\leq$	1
(6)			1		1				$\leq$	1
(7)			1					1	$\leq$	1
(8)				1		1			$\leq$	1
(9)				1	1		1		$\leq$	1

Table 5: The linear program constructed using the independent sets $I(1), \ldots, I(9)$ of the graph G in Example 2.

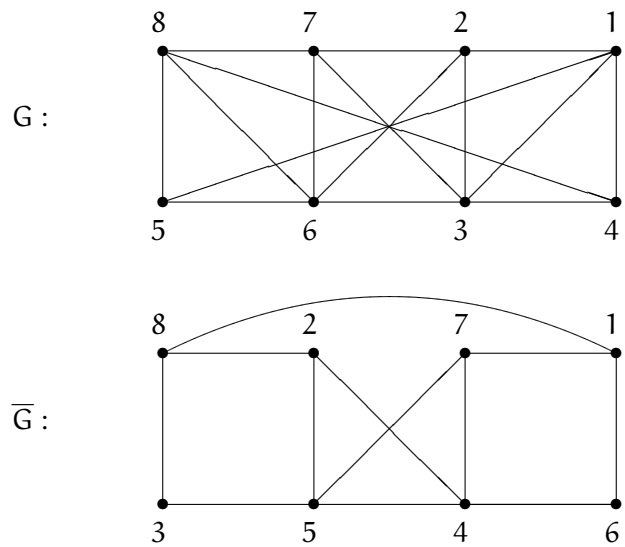

Figure 1: Graphical representations of the graph G and its complement $\overline{G}$ in Example 2.

2 The edge auxiliary graph

To a given finite simple graph $G = (V, E)$ we assign a new graph $\Gamma = (W, F)$. We call Γ the edge auxiliary graph of G. The nodes of Γ are the edges of G, that is, $W = E$. Let us consider two distinct nodes

$$w_1 = \{u_1, v_1\}, \quad w_2 = \{u_2, v_2\} \tag{1}$$

of Γ. Here the unordered pairs $\{u_1, v_1\}$, $\{u_2, v_2\}$ are edges of G. We construct the subset $X = \{u_1, v_1, u_2, v_2\}$ of V. Since the nodes w_1, w_2 are not equal, the set X has either 3 or 4 elements. If the subgraph induced by X in G is a clique in G, then we say that X is a qualifying subset of V. The nodes (1) are adjacent in Γ if the associated subset X is qualifying.

We are interested in the interplay between the cliques in the graphs G and Γ. If a subset U of V is the set of nodes of a k-clique in G, then the unordered pairs $\{u, v\}$, $u, v \in U$ are the nodes of an s-clique in Γ, where $s = k(k-1)/2$. In other words cliques in G give rise to cliques in Γ. Cliques in Γ also give rise to cliques in G. Let

$$w_1 = \{u_1, v_1\}, \ldots, w_s = \{u_s, v_s\} \tag{2}$$

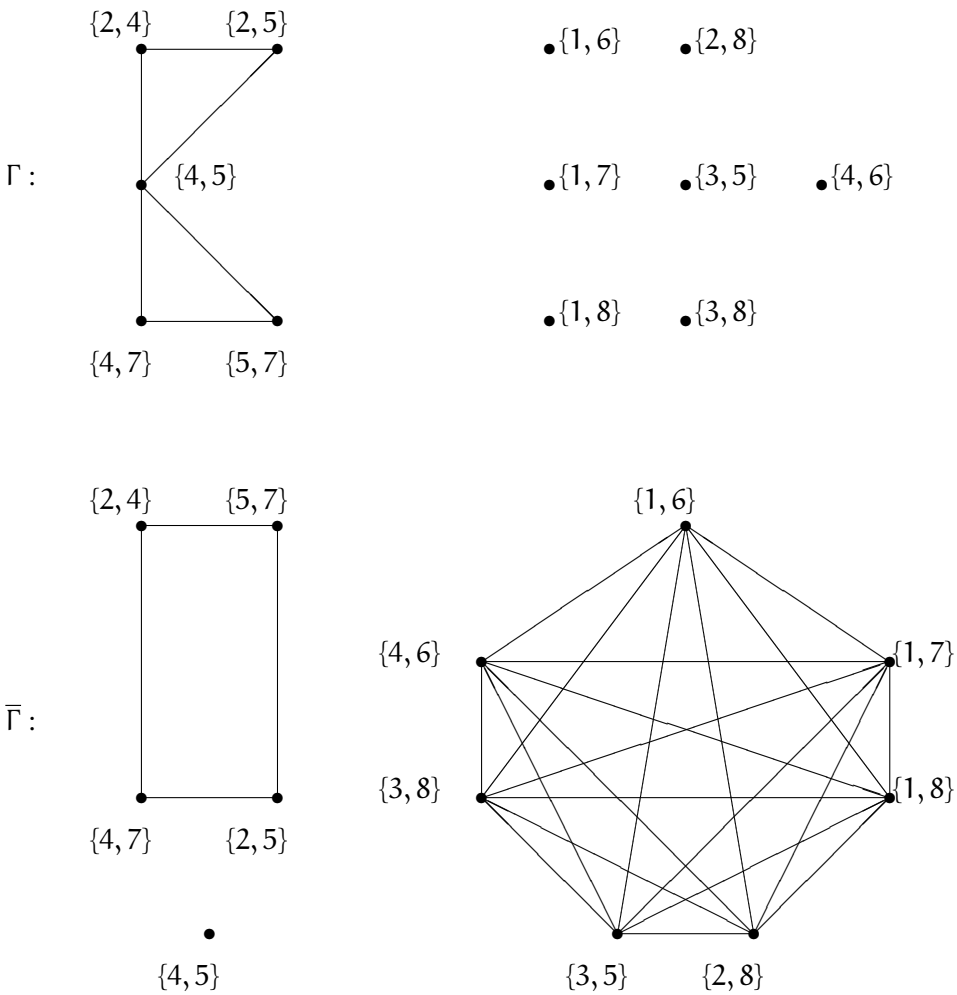

Figure 2: Graphical representations of the graph Γ and its complement $\overline{\Gamma}$ in Example 2. Each node of the graph on the right in $\overline{\Gamma}$ is adjacent to each node of the graph on the left. In order to avoid an overly cluttered picture we omitted these 35 edges.

parameter of G	number of nodes of G	number of nodes of $\overline{\Gamma}$	number of colors	estimate for $\omega(G)$
3	27	162	17	5
4	64	720	44	9
5	125	2 250	91	14
6	216	5 670	165	21
7	343	12 348	272	29
8	512	24 192	417	38
9	729	43 740	608	48
10	1 000	74 250	850	59
11	1 331	119 790	1 148	72
12	1 728	185 328	1 510	85
13	2 197	276 822	1 940	100

Table 6: Monoton matrices

be vertices of an s-clique in Γ and let

$$x_1, \ldots, x_k \tag{3}$$

be all the distinct elements among $u_1, v_1, \ldots, u_s, v_s$.

Lemma 1 *Using the notation introduced above the nodes* $x_1, \ldots, x_k$ *of* G *are the nodes of a* k*-clique in* G.

Proof. We show that the unordered pair $\{x_i, x_j\}$ is a edge of the graph G for each i, j, $1 \le i < j \le k$. As nodes of Γ on the list (2) are nodes of a clique in Γ it follows that there is an x_p on the list (3) such that the unordered pair $w_1' = \{x_i, x_p\}$ is an edge of G. Similarly, there is an x_q on the list (3) such that the unordered pair $w_2' = \{x_j, x_q\}$ is an edge of G. Using the fact that the nodes of Γ on the list (2) are the nodes of a clique in Γ we can draw the conclusion that the subset $X = \{x_i, x_p, x_j, x_q\}$ associated with w_1' and w_2' is qualifying. Consequently, the unordered pair $\{x_i, x_j\}$ is an edge of a clique in the graph G. (It is an edge of either a 3-clique or a 4-clique.) In particular, the unordered pair $\{x_i, x_j\}$ is an edge of the graph G. □

In the next lines we summarize the work flow of computing an upper bound for the clique number of a given finite simple graph.

(1) Using the given graph G we construct its complement graph $\overline{G}$.

parameter of G	number of nodes of G	number of nodes of $\overline{\Gamma}$	number of colors	estimate for $\omega(\mathsf{G})$
3	8	19	5	2
4	16	63	9	4
5	32	191	18	6
6	64	543	34	10
7	128	1 471	66	17
8	256	3 839	130	30
9	512	9 724	258	53
10	1024	24 063	514	96
11	2048	58 367	1 026	175

Table 7: Deletion error detecting codes

parameter of G	number of nodes of G	number of nodes of $\overline{\Gamma}$	number of colors	estimate for $\omega(\mathsf{G})$
6	15	60	20	5
7	35	210	35	8
8	70	560	56	11
9	126	1 260	84	21
10	210	2 520	120	30
11	330	4 620	165	41
12	495	7 920	220	55
13	715	12 870	286	71
14	1 001	20 020	364	91
15	1 365	30 030	455	113
16	1 820	43 680	560	140
17	2 380	61 880	680	170

Table 8: Johnson codes

(2) Using $\overline{G}$ we construct its edge auxiliary graph Γ.

(3) From Γ we construct its complement graph $\overline{\Gamma}$.

(4) We color the nodes of $\overline{\Gamma}$ legally. The color classes $C(1), \ldots, C(s)$ of the nodes of $\overline{\Gamma}$ are cliques in Γ.

(5) Using the cliques $C(1), \ldots, C(s)$ in Γ we construct cliques $I(1), \ldots, I(s)$ in $\overline{G}$. (We apply Lemma 1.) The cliques $I(1), \ldots, I(s)$ in $\overline{G}$ are independent sets in G.

(6) Using the independent sets $I(1), \ldots, I(s)$ in G we construct a linear program P.

(7) Solving the linear program P gives an upper bound for $\omega(G)$.

3 A small size toy example

In this section we work out a small example in details. Our intension is to illustrate the concepts and arguments we have seen earlier.

Example 2 *Let us consider the finite simple graph* $G = (V, E)$ *defined by its adjacency matrix in Table 1. The graph* G *has* 8 *vertices and* 16 *edges. The vertices are denoted by* $1, \ldots, 8$*, that is,* $V = \{1, \ldots, 8\}$*. A possible geometric representation of* G *can be seen in Figure 1.*

An inspection shows that $\omega(G) = 4$. We pretend that we are not aware of this fact and with full seriousness we carry out a procedure to establish an upper estimate for $\omega(G)$.

Table 1 shows the adjacency matrix of the given graph G. This is adjacency matrix is on the left. We constructed the adjacency matrix of the graph $\overline{G}$. This can be seen on the right. Possible geometric representations of these graphs are in Figure 1. Using $\overline{G}$ we constructed its edge auxiliary graph Γ. Table 2 shows the adjacency matrices of the edge auxiliary graph Γ of the graph $\overline{G}$ and its complement $\overline{\Gamma}$. Possible geometric representations of these graphs are depicted by Figure 2.

We used the simplest greedy coloring procedure to construct a legal coloring of the nodes of $\overline{\Gamma}$. Table 3 summarizes the procedure. The first column contains the nodes of $\overline{\Gamma}$ and the last column holds the colors of the nodes. The color classes $C(1), \ldots, C(9)$ of the nodes of $\overline{\Gamma}$ are nodes of cliques in Γ and are listed in

Table 4. This table also lists the independent sets $I(1), \ldots, I(9)$ corresponding to the cliques $C(1), \ldots, C(9)$ in the given graph G.

We used these independent sets to set up a linear program. The linear program is given by Table 5.

4 Numerical experiments

The edge auxiliary graph Γ, we construct from the complement $\overline{G}$ of the given graph G, has as many vertices as many edges $\overline{G}$ has. Then we locate a legal coloring of the nodes of $\overline{\Gamma}$ the complement of the auxiliary graph Γ. We carry out numerical experiments to see whether greedy coloring of such large graphs gives results that are useful in practical setting.

We use three infinite families of graphs as test cases. All of them are related to coding theory. The graphs associated with monotonic matrices are described in details in [6] and [7]. The graphs connected to deletion error detecting codes are presented in [5]. Finally the reader can find further material about the so-called Johnson codes in [3].

Table 6 contains the result of our computation related to the graphs associated with monotonic matrices. The first column contains the parameter of the graph G we work with. In the second column one can find the number of the nodes of G. The third column holds the number vertices of $\overline{\Gamma}$ which is equal to the number of edges of $\overline{G}$. The fourth column lists the number of colors used for legally coloring the nodes of $\overline{\Gamma}$. Finally, the last column shows the upper estimate we get for $\omega(G)$.

Tables 7 and 8 can be interpreted in an analogous way. We may conclude that the procedure we proposed to set up a linear program to estimate the clique number of a given graph can be carried out safely in connection with relatively large graphs that occur in practical clique search problems.

Acknowledgements

I would like to express my thanks for the anonymous referees.

References

[1] I. M. Bomze, M. Budinich, P. M. Pardalos, M. Pelillo, The Maximum Clique Problem, Handbook of Combinatorial Optimization Vol. 4, Kluwer Academic Publisher, 1999. ⇒123

[2] M. R. Garey, D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-completeness*, Freeman, New York, 2003. ⇒123

[3] J. Hasselberg, P. M. Pardalos, G. Vairaktarakis, Test case generators and computational results for the maximum clique problem, *Journal of Global Optimization* **3** (1993) 463–482. ⇒132

[4] C. H. Papadimitriou, *Computational Complexity*, Addison-Wesley Publishing Company, Inc., Reading, MA 1994. ⇒123

[5] N. J. A. Sloane, *Challenge Problems: Independent Sets in Graphs.* https://oeis.org/A265032/a265032.html ⇒132

[6] S. Szabó, Monotonic matrices and clique search in graphs, *Annales Univ. Sci. Budapest., Sect. Computatorica* **41** (2013), 307–322. ⇒132

[7] E. W. Weisstein, Monotonic Matrix, In: *MathWorld–A Wolfram Web Resource.* http://mathworld.wolfram.com/MonotonicMatrix.html ⇒132

Received: April 1, 2021 • Revised: May 16, 2021