An Enhanced Infra-Chromatic Bound for the Maximum Clique Problem

Pablo San Segundo(✉), Jorge Artieda, Rafael Leon, and Cristobal Tapia

Center for Automation and Robotics (CAR),
Polytechnic University of Madrid (UPM),
C/ Jose Gutiérrez Abascal, 2, 28006 Madrid, Spain
pablo.sansegundo@upm.es

Abstract. There has been a rising interest in experimental exact algorithms for the maximum clique problem because the gap between the expected theoretical performance and the reported results in practice is becoming surprisingly large. One reason for this is the family of bounding functions denoted as *infra-chromatic* because they produce bounds which can be lower than the chromatic number of the bounded subgraph. In this paper we describe a way to enhance exact solvers with an additional infra-chromatic bounding function and report performance over a number of graphs from well known data sets. Moreover, the reported results show that the new enhanced procedure significantly outperforms state-of-the-art.

Keywords: Infra-chromatic · Clique · Approximate-coloring · Branch-and-bound · Combinatorial optimization · Search

1 Introduction

Given a simple undirected graph $G = (V, E)$, with n vertices and m edges, $N(v)$ refers to the neighbor set of vertices adjacent to a vertex $v \in V$ and $G[U]$ to the subgraph induced by a vertex set $U \subseteq V$. A complete subgraph (or clique) is an induced subgraph such that all its vertices are pairwise adjacent, and the maximum clique problem (MCP) consists in finding a clique of maximum cardinality. The MCP is a theoretical deeply studied NP-hard problem which has also found many applications [1]. Examples of clique search appear in network analysis, coding theory, fingerprint matching, bioinformatics [2], robotics [3, 4] and computer vision [5].

In the last decade there has been rising interest in the exact solving of the MCP. At present, almost all successful algorithms are branch-and-bound and use a greedy vertex coloring procedure as bound [6–15]. Interestingly, the reports provided by these practical algorithms average a much better performance than is to be expected by the theoretical algorithms alone. A brief overview of theoretical results now follows:

P.M. Pardalos et al. (Eds.): MOD 2016, LNCS 10122, pp. 306–316, 2016.
DOI: 10.1007/978-3-319-51469-7_26

- The MCP is *NP*-hard and can not even be approximated in polynomial time within a factor of $|V|^{1/3-\varepsilon}$, for any $\varepsilon > 0$(unless $P = NP$) [16].
- There can be as much as $3^{n/3}$ distinct maximal cliques to enumerate in a graph of size n [17], which is an upper bound for all enumerative solvers.
- Branch-and-bound solvers do not need to explore all the maximal cliques. An important number of them may be discarded when, during construction, it is shown that they cannot improve the current incumbent clique. In [18], the worst case running time was set to $O(2^{n/3})$ and, at present, the value of this bound is $O(2^{n/4})$ [19], in both cases with exponential space consumption. We note that these algorithms have not been implemented in practice and literature provides no experimental evaluation.
- An interesting recent work is [20] which is concerned with the family of branch-and-bound solvers which use coloring as bounding function; these are also the reference algorithms for this work. [20] sets $\Omega(2^{n/5})$ as lower bound for this family of solvers.

The latter result is especially relevant because it is related to state-of-the-art exact algorithms and, as far as we are aware, the gap between the good performances shown by the experimental algorithms and the theoretical result remains to be explained. Motivated by this fact, this work contributes with an empirical study of cutting-edge *infra-chromatic* bounding functions which have recently been described for the MCP. These functions are able to bind the clique number of a given subproblem below its chromatic number by additional computational effort. More precisely, this implies that the theoretical $\Omega(2^{n/5})$ bound does not hold for these algorithms.

Leading infra-chromatic experimental algorithms for the MCP at present are MaxCLQ [14], and its improved variant IncMaxCLQ [15], as well as bit-parallel BBMCX [11] — see recent comparison surveys [21, 22]. The contribution of this work is experimental. First we describe a way to enhance BBMCX with an additional infra-chromatic bounding function closely related to the one described in MaxCLQ. This bounding function can also be applied to earlier BBMC variants [8–10]. The paper reports performances of two enhanced variants (including BBMCX), together with IncMaxCLQ.

The remaining part of this work is structured in the following way: Sects. 2 and 3 cover prior related algorithms. The new bounding function for exact maximum clique is described in Sect. 4 and validated empirically in Sect. 5. Finally, Sect. 6 briefly discusses the reported results and Sect. 7 presents some conclusions.

2 Related Bounding Functions for the MCP

In the last decade, greedy sequential vertex coloring (SEQ) has proved to be a good bounding function for exact branch-and-bound solvers for the MCP. SEQ is an $O(n^2)$ worst-case bounded procedure which iteratively assigns the lowest possible color

number to each vertex that is consistent with the current partial coloring. The first maximum clique algorithm of this type described in literature is Fahle's [6]. Since then, much research has been devoted to improve the basic SEQ bounding function.

We denote by $C(G) = \{C_1, C_2, \ldots, C_k\}$ a k-coloring of graph G, where C_i refers to the (color) set which contains vertices assigned color number i. For a given k-coloring $C(G)$, maximum clique solvers use a *recoloring* bounding function to lower the color number k of a vertex $v_1 \in C_k$ by a swap movement with two already colored vertices $v_2 \in C_i$ and $v_3 \in C_j$ where $i<j<k$. A successful recoloring leads to the new coloring $v_1 \in C_i$ and $v_2, v_3 \in C_j$. In practice, recoloring can be applied to every vertex of the SEQ coloring, as described originally for algorithm MCS [12], or relaxed to a particular subset, as in BBMCR [9].

A bounding function that could produce bounds below the chromatic number of the bounded subproblem was first described in algorithm MaxCLQ [14], and later improved in IncMaxCLQ [15]. It was defined in terms of a reduction of the MCP over a colored graph to a partial maximum satisfiability problem (PMAX-SAT) as follows: (I) Each vertex of the graph maps to a logical variable. (II) Each PMAX-SAT hard clause consists of two negated literals and represents a non-adjacency relation between the pair of corresponding vertices. (III) Each PMAX-SAT soft clause represents a distinct color set and contains the corresponding positive literals that map to vertices of the set.

Figure 1 provides a simple example of the reduction when applied to an odd-cycle:

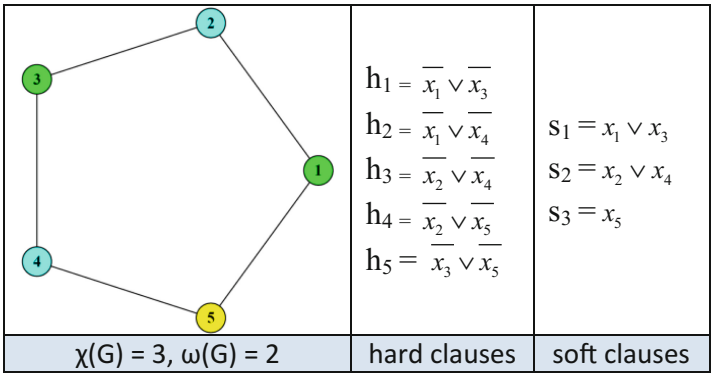

Fig. 1. An example of an MCP reduction to PMAX-SAT. (Color figure online)

The PMAX-SAT problem's equivalence to the MCP consists in finding an assignment of literals which satisfy the maximum number of soft clauses (the color sets) and all hard clauses (the graph structure) — see [14, 15] for a more detailed description. The cleverness of this reduction for branch-and-bound exact MCP solvers

is that well-known pruning strategies for PMAX_SAT may be used as bounding functions for the MCP. The main inference is based on propagating literal assignments of *unit* soft clauses (singleton color sets) until an empty clause is reached. When this occurs, the soft clauses (color sets) that make part of the reasoning are said to be *inconsistent*. Intuitively, the subgraph induced by an inconsistent set of k colors cannot contain a clique of size k or bigger, so the bound provided by the number of colors in the original coloring may be decremented by one *for each inconsistent set of colors found*.

In the example of Fig. 1, one such inconsistent set of soft clauses is $\{s_1, s_2, s_3\}$, which can be obtained by the following inference: (I) $x_5 \leftarrow TRUE$ since s_3 is a unit clause. (II) Propagating $x_5 \leftarrow TRUE$ value leads to $s_2 = x_4$, according to hard clause h_4, and $s_1 = x_1$, according to hard clause h_5. (III) $x_4 \leftarrow TRUE$, since s_2 is a unit clause. IV) Propagating $x_4 \leftarrow TRUE$ leads to the empty clause $s_1 = \emptyset$, according to the hard clause h_2. Consequently, the original color bound of the graph, 3, can be reduced to 2. Note that the chromatic number of the graph $\chi(G) = 3$ is higher than the new bound.

The term *infra-chromatic* describing bounding functions for the MCP was first employed in the bit-parallel BBMCX algorithm. There, the authors propose an inference procedure in terms of color sets which integrates well with the bitstring encoding and branching of the family of BBMC algorithms [8–11]. More precisely, BBMCX looks for triplets of color sets $\{C_x, C_y, C_z\}$ such that $|C_x| = 1$ and no triangles exist in the induced graph $G[C_x \cup C_y \cup C_z]$. The reported results show that this particular case of inconsistency can be computed very efficiently achieving a very good tradeoff between pruning ability and computational effort.

3 The Prior Infra-Chromatic Procedure

Rising interest in exact maximum clique search combines exhaustive enumeration of maximal cliques with a bounding function based on approximate coloring. A step of the search (typically a recursive call to the algorithm) branches on a vertex to enlarge a clique, and a branch of the search tree corresponds to a maximal clique. The solution to the MCP is any leaf node of maximum cardinality.

In this work we borrow the notation for sets employed in BBMCX [11]. More specifically:

- S denotes the clique to be enlarged at any point during search.
- S_{max} is the largest clique found at any point during search.
- U denotes the set of vertices of the current subproblem.
- U_v denotes the set of vertices of the child subproblem resulting from branching on vertex $v \in U$.

- $G_v = G[U_v]$ refers to the induced graph subproblem which results from branching on vertex v.
- F is a set of (forbidden) color numbers employed in the infra-chromatic bounding function described in BBMCX.

Of interest to this work is the bounding function called for each subproblem U during search. Algorithm 1 outlines the bounding procedure which roughly corresponds with the one described in BBMCX. Procedure UPPERBOUND computes the SEQ coloring of the subproblem $G[U_v]$ in the main loop (steps 1 to 12), by enlarging one color set at a time. Once all color sets below a certain pruning threshold k_{min} have been constructed, each selected vertex goes through the infra-chromatic filter INFRACHROM_I described in Algorithm 2.

Algorithm 1.Outline of the BBMCX [11] bounding function.

INPUT: A graph $G[U_v]$ and a color threshold k_{min}

UPPERBOUND (U_v, $k_{min} = |S_{max}| - |S| + 1$)

Initial step: $U \leftarrow U_v$, $F \leftarrow \varnothing$, $k \leftarrow 1$, $C \leftarrow \varnothing$

1. **repeat until** $U = \varnothing$
2. $C_k \leftarrow U$
3. **repeat until** all vertices in C_k have been selected
4. select the first possible vertex *v* from C_k
5. result← FAIL
6. **if** $k \geq k_{min}$ **then** result←INFRACHROM_I (*v*, k_{min}, $C_1, C_2, \cdots, C_{k_{min}-1}, C_k$,*F*)
7. **if** result = FAIL **then** $C_k \leftarrow C_k \setminus N(v)$
8. $U \leftarrow U \setminus \{v\}$
9. **endrepeat**
10. $C \leftarrow C \cup C_k$
11. $k := k + 1$
12. **endrepeat**
13 .**return** $|C|$

The filter function looks for pairs of distinct color classes (i, j) such that vertex v is adjacent to exactly just one vertex w in color i and color j contains no common neighbors to both v and w. If the filter succeeds (and returns SUCCESS) the vertex is removed from the original coloring and the other two inconsistent color sets are tagged as forbidden, so that they do not take part in further inferences. We refer the reader to [11] for additional design and implementation details.

Algorithm 2. Outline of the infra-chromatic bounding function called by Algorithm 1.

INFRACHROM_I (v, k_{min}, $C_1, C_2, \cdots, C_{k_{min}-1}, C_k$, F) //F is a list of forbidden colors, $v \in C_k$

1. find any color class i not in F such that it has only one neighbor w in common with v
2. find any color class $j \neq i$ not in F such that it contains no common neighbors to both w and v
3. **if** steps 1 and 2 are successful
4. add color labels i and j to F
5. remove v from C_k
6. **return** SUCCESS
7. **repeat** steps 1 to 6 with all feasible colors
8. **return** FAIL

4 The Enhanced Infra-Chromatic Procedure

We enhance the previously described bounding function with another infra-chromatic bound which employs unit clause propagation described in PMAX-SAT based solvers. Algorithms 3 and 4 describe the general outline of the new enhanced function. The coloring C obtained after applying INFRACHROM_I to SEQ is now filtered again by new INFRACHROM_II. The function searches for inconsistent color sets using unit clause inferences and returns the number of conflicts found. As explained in the introductory section, the resulting new bound is the difference between the previous bound (the size of the coloring) and the number of conflicts outputted by INFRACHROM_II.

Algorithm 3. Outline of the enhanced Algorithm 1.

UPPERBOUND (U_v, $k_{min} = |S_{max}| - |S| + 1$) //$k_{min}$ is a color number threshold

//... same steps as Algorithm 1

12. **endrepeat**
13. **if** $|C| \geq k_{\min}$ **then**
14. *num_conf*←INFRACHROM_II(C, F)
15. **endif**
16. **return** $|C|$ - *num_conf*

In the current implementation of INFRACHROM_II, the forbidden color set F and the color and neighbor sets are encoded as bitsets. The main inference is the set intersection operation in step 7 which benefits from bit-masking, as well as the initial step 1. The procedure runs in $O(|V|^2)$ in the worst-case. In the current implementation, we obtain the best performance when color sets are examined in non-increasing order. This strategy tends to find conflicts earlier because the highest color sets of the SEQ

coloring are expected to have a small size. We note that it is easy to envisage more greedy variants of Algorithm 4 by filtering out color sets from the reasoning with additional constraints. It is also worth noting that the integration of INFRACHROM_II in the prior bounding function (see Algorithm 3) is not restricted to BBMCX, and may also make part of other exact approximate-color exact solvers such as MCS, BBMC, BBMCR and other published BBMC variants.

Algorithm 4. Outline of the infra-chromatic procedure called by Algorithm 3.

INPUT: A vertex coloring *C* and a list *F* of color numbers
INFRACHROM_II (*C*, *F*)
Initial step 1: Remove all color sets with labels in *F* from *C*
Initial step 2: *UC* ← all singleton color sets in *C*, *num_conf* ← 0
1. **repeat until** $UC = \varnothing$
2. $UC' \leftarrow \varnothing$
3. move the first singleton color set in UC to UC'
4. **repeat until** $UC' = \varnothing$
5. select (and remove) a singleton color set $C_k = \{v\}$ from UC'
6. tag C_k as inconsistent
7. **for** every $C_{j \neq k}$ **do** $C_j \leftarrow C_j \cap N(v)$
8. **for** every $C_{j \neq k}$
9. **if** $|C_{j \neq k}| = 1$ **then** $UC' \leftarrow UC' \cup C_j$
10. **else if** $|C_{j \neq k}| = \varnothing$ **then**
11. *num_conf* ← *num_conf* + 1
12. remove C_j and all sets tagged as inconsistent from UC
13. restore all color sets not tagged as inconsistent
14. **goto** step 1
15. **endif**
16. **endfor**
17. **endrepeat**
18. **endrepeat**
19. **return** *num_conf*

5 Experiments

To evaluate the new proposed infra-chromatic bounding function we have considered the following algorithms in this research:

1. BBMC [8, 9]: The original bit-parallel algorithm for the MCP, which uses only the SEQ color bound.
2. BBMC+*I*: BBMC enhanced with the new PMAX-SAT based infra-chromatic bound described in Algorithm 3.

3. BBMCR [9]+*I*: The BBMC recoloring variant enhanced with the new infra-chromatic bound.
4. BBMCX [11]+*I:* The BBMC infra-chromatic variant enhanced with the new infra-chromatic bound.
5. IncMaxCLQ [15]: The latest PMAX-SAT based algorithm, as provided by its main developer.

All the algorithms were tested over a big number of structured graphs from public datasets which may be found in other maximum clique reports elsewhere. The majority of structured instances were presented at the Second DIMACS Implementation Challenge[1]. The rest are taken from the BHOSHLIB benchmark[2] (more specifically, the *frb30-15* collection). The algorithms were also run against uniform random graphs, with sizes ranging from 150 to 15000 vertices and varying densities.

The hardware used in the experiments was a 20 core XEON with 64 GB of RAM running a Linux OS and all algorithms considered were run on a single core of this machine. With respect to initial configurations, the enhanced BBMC infra-chromatic variants use the more recent preprocessing ideas, that is, the initial sorting of vertices described in IncMaxCLQ, and a strong initial solution computed by a state-of-the-art heuristic, as recommended in [13].

Due to space constraints we report in this manuscript the performance of algorithms BBMCR+*I,* BBMCX+*I* and IncMaxCLQ against 37 representative structured graphs — solved in more than 0.01 s by at least one algorithm.

6 Discussion

Table 1 reports the performance of the enhanced algorithms BBMCR+*I,* BBMCX+*I* as well as IncMaxCLQ. Interestingly, out of the 37 instances reported, BBMCR+*I* is faster than BBMCX+*I* in 25 of them. This would indicate that recoloring (BBMCR), followed by a single infra-chromatic filter, is more appropriate than the double infra-chromatic filter of BBMCX+*I*. This is corroborated by the number of steps taken by both algorithms.

In relation to IncMaxCLQ, the enhanced algorithms perform better in 27 of the 37 instances. Worth noting is that IncMaxCLQ clearly outperforms the latter in *keller5* and *C250.9*. Overall, IncMaxCLQ produces smaller search trees but with higher computational cost, the tradeoff being favorable only in very dense graphs (that is, $p \geq 0.9$) with the exception of *keller5*. It is also worth noting that the number of steps taken by both enhanced algorithms are much less, on average, than those reported for BBMCX in [11].

[1] http://cs.hbg.psu.edu/txn131/clique.html.

[2] http://www.nlsde.buaa.edu.cn/~kexu/benchmarks/graph-benchmarks.htm.

Table 1. A comparison of different infra-chromatic exact maximum clique algorithms. Time is measured in seconds with precision of milliseconds. Header *Inc/BBMC* compares performance as a ratio. Header *p* refers to uniform density and header ω is the size of the maximum clique. Cells in bold show the best times in each row. A *step* is a recursive call in the BBMC enhanced algorithms.

			IncMaxCLQ [15]		BBMCR [9]+/		BBMCX [11]+/		Inc/BBMC
	p	ω	steps	time	steps	time	steps	time	time
C250.9	0.90	44	734186	**228**	40940184	832	34457025	1144	0.3
MANN_a27	0.99	126	290	**0.266**	3947	0.330	2749	0.318	0.8
MANN_a45	0.99	345	21595	101	40024	**39.6**	39944	41.4	2.5
brock200_1	0.75	21	2700	0.482	23006	**0.250**	29960	0.357	1.9
brock400_1	0.75	27	890529	**182**	11695876	233	16761126	358	0.8
brock400_2	0.75	29	711295	139	3594552	**84.4**	5372259	132	1.7
brock400_3	0.75	31	887764	170	803747	**25.0**	1339642	40.9	6.8
brock400_4	0.75	33	640279	129	575792	**19.9**	575787	20.2	6.5
brock800_1	0.65	23	39290054	7846	199341746	6381	134188081	**4295**	1.8
brock800_2	0.65	24	49803966	10011	160679838	5380	107980185	**3586**	2.8
brock800_3	0.65	25	17522630	3759	52854672	**2012**	79775569	3037	1.9
brock800_4	0.65	26	24600360	4768	45605176	**1669**	68040462	2532	2.9
dsjc1000.5	0.50	15	1562003	197	5323240	**157**	7226392	211	1.3
dsjc500.5	0.50	13	19303	2.63	93413	**1.26**	123963	1.66	2.1
frb30-15-4	0.82	30	22	0.213	1	**<0.001**	1	**<0.001**	213
frb30-15-5	0.82	30	9	0.163	1	**<0.001**	1	**<0.001**	163
gen400_p0.9_55	0.90	55	320	1.05	1	**<0.001**	1	**<0.001**	1052
gen400_p0.9_65	0.90	65	75	0.285	2	**<0.001**	2	**<0.001**	285
gen400_p0.9_75	0.90	75	160	0.238	1333	**0.063**	1314	0.073	3.8
hamming10-2	0.99	512	0	25.0	1	**<0.001**	1	**<0.001**	24992
keller5	0.75	27	249409	**127**	90179996	2122	95938710	2667	0.1
p_hat1000-1	0.25	10	1700	0.503	17102	**0.301**	20948	0.361	1.7
p_hat1000-2	0.49	46	51658	**43.2**	843212	54.1	1237187	93.2	0.8
p_hat1500-1	0.25	12	56404	4.64	76509	**2.37**	88973	3.18	2.0
p_hat300-3	0.74	36	506	**0.313**	11205	0.359	15797	0.531	0.9
p_hat500-2	0.51	36	254	0.165	2007	**0.093**	2945	0.148	1.8
p_hat500-3	0.75	50	28338	**19.6**	605339	27.7	833148	44.7	0.7
p_hat700-2	0.50	44	613	0.826	16688	**0.815**	22152	1.42	1.0
p_hat700-3	0.75	62	247906	**241**	5474929	408	8241315	718	0.6
san1000	0.50	15	646	0.421	15210	**0.412**	9981	0.750	1.0
san400_0.7_2	0.70	30	736	0.357	1	**<0.001**	1	**<0.001**	357
san400_0.7_3	0.70	22	7364	2.12	1	**<0.001**	1	**<0.001**	2121
san400_0.9_1	0.90	100	99	0.223	1	**<0.001**	2	**<0.001**	223
sanr200_0.7	0.70	18	1366	**0.193**	15210	1.003	9981	0.750	0.3
sanr200_0.9	0.90	42	8029	**2.28**	320666	6.33	305083	9.19	0.4
sanr400_0.5	0.50	13	5041	0.491	13947	**0.265**	18246	0.336	1.9
sanr400_0.7	0.70	21	477635	90.2	4749393	**73.6**	6450260	107	1.2

7 Conclusions

This work describes a simple way to enhance a leading bit-parallel infra-chromatic solver for the maximum clique problem with a PMAX-SAT based infra-chromatic filter, and compares the performance of two enhanced published solvers with another state-of-the-art algorithm. The contribution is a step forward in the search of an adequate explanation for the gap between theoretical results and the excellent performance shown by experimental algorithms for this problem. Moreover, the enhanced algorithms show some of the best times published, to the best of our knowledge, for a number of structured graphs from well known public data sets.

Acknowledgments. This work is funded by the Spanish Ministry of Economy and Competitiveness (grant NAVEGASE: DPI 2014-53525-C3-1-R).

References

1. Bomze, I., Budinich, M., Pardalos, P., Pelillo, M.: The maximum clique problem. In: Du, D.-Z., Pardalos, P.M. (eds.) Handbook of Combinatorial Optimization, vol. 4, pp. 1–74. Springer, New York (1999)
2. Butenko, S., Chaovalitwongse, W., Pardalos, P. (eds.): Clustering Challenges in Biological Networks. World Scientific, Singapore (2009)
3. San Segundo, P., Rodriguez-Losada, P., Matia, D., Galan, R.: Fast exact feature based data correspondence search with an efficient bit-parallel MCP solver. Appl. Intell. **32**(3), 311–329 (2010)
4. San Segundo, P., Rodriguez-Losada, D.: Robust global feature based data association with a sparse bit optimized maximum clique algorithm. IEEE Trans. Robot. **99**, 1–7 (2013)
5. San Segundo, P., Artieda, J.: A novel clique formulation for the visual feature matching problem. Appl. Intell. **43**(2), 325–342 (2015)
6. Fahle, T.: Simple and fast: improving a branch-and-bound algorithm for maximum clique. In: Möhring, R., Raman, R. (eds.) ESA 2002. LNCS, vol. 2461, pp. 485–498. Springer, Heidelberg (2002). doi:10.1007/3-540-45749-6_44
7. Tomita, E., Seki, T.: An efficient branch-and-bound algorithm for finding a maximum clique. In: Calude, C.S., Dinneen, M.J., Vajnovszki, V. (eds.) DMTCS 2003. LNCS, vol. 2731, pp. 278–289. Springer, Heidelberg (2003). doi:10.1007/3-540-45066-1_22
8. Segundo, S., Rodriguez-Losada, D, Jimenez, A.: An exact bit-parallel algorithm for the maximum clique problem. Comput. Oper. Res. **38**(2), 571–581 (2011)
9. San Segundo, P., Matia, F., Rodriguez-Losada, D., Hernando, M.: An improved bit parallel exact maximum clique algorithm. Optim. Lett. **7**(3), 467–479 (2013)
10. San Segundo, P., Tapia, C.: Relaxed approximate coloring in exact maximum clique search. Comput. Oper. Res. **44**, 185–192 (2014)
11. San Segundo, P., Nikolaev, A., Batsyn, M.: Infra-chromatic bound for exact maximum clique search. Comput. Oper. Res. **64**, 293–303 (2015)
12. Tomita, E., Sutani, Y., Higashi, T., Takahashi, S., Wakatsuki, M.: A simple and faster branch-and-bound algorithm for finding a maximum clique. In: Rahman, Md.S, Fujita, S. (eds.) WALCOM 2010. LNCS, vol. 5942, pp. 191–203. Springer, Heidelberg (2010). doi:10.1007/978-3-642-11440-3_18
13. Batsyn, M., Goldengorin, B., Maslov, E., Pardalos, P.: Improvements to MCS algorithm for the maximum clique problem. J. Comb. Optim. **27**, 397–416 (2014)
14. Li, C.-M., Quan, Z.: Combining graph structure exploitation and propositional reasoning for the maximum clique problem. In: Proceedings of ICTAI, pp. 344–351 (2010)
15. Li, C.-M., Fang, Z., Xu, K.: Combining MaxSAT reasoning and incremental upper bound for the maximum clique problem. In: Proceedings of ICTAI, pp. 939–946 (2013)
16. Bellare, M., Goldreich, O., Sudan, M.: Free bits, PCPs and nonapproximability — towards tight results. In: 1995 Proceedings of 36th Annual Symposium on Foundations of Computer Science, pp. 422–431. IEEE (1995)
17. Moon, J., Moser, L.: On cliques in graphs. Isr. J. Math. **3**(1), 23–28 (1965)
18. Tarjan, R.E., Trojanowski, A.E.: Finding a maximum independent set. Technical report, Computer Science Department, School of Humanities and Sciences, Stanford University, Stanford, CA, USA (1976)

19. Robson, J.M.: Finding a maximum independent set in time $O(2^{n/4})$. Technical report 1251-01, LaBRI, Université de Bordeaux I (2001)
20. Lavnikevich, N.: On the complexity of maximum clique algorithms: usage of coloring heuristics leads to the $\Omega(2^{n/5})$ algorithm running time lower bound (2013)
21. Prosser, P.: Exact algorithms for maximum clique: a computational study. Algorithms **5**(4), 545–587 (2012)
22. Wu, Q., Hao, J.K.: A review on algorithms for maximum clique problems. Eur. J. Oper. Res. **242**(3), 693–709 (2015)