



# **A Fast Color Based Branch and Bound Algorithm for Maximum Clique Problem**

Krishna Kumar Singh<sup>a</sup>, K N M Satyanarayana<sup>b</sup>

Asst Professor, Dept. of CSE, RGUKT-Nuzvid, AP, India<sup>a</sup>

B. Tech Final Year Student, Dept. of CSE, RGUKT-Nuzvid, AP, India<sup>b</sup>

**ABSTRACT:** A clique is a sub graph in which all pairs of vertices are mutually adjacent. Finding maximum clique in a graph is a NP-complete problem. This paper proposes an efficient color based heuristic branch and bound approach for finding maximum clique. It is assumed that the target (maximum clique size) is known and taken as input to the algorithm. At each recursive step, the algorithm finds promising set vertices of particular color class which are close to the remaining target at depth  $d$ . The obtained promising set of vertices is used for further expansion to get adjacent list. A color class contains a set of maximal independent set. It is not hard to believe that at the depth, the maximum clique most likely would include one of the vertices of maximal independent set. Additionally some pruning strategies are used to abort smaller size of clique to be explored, e.g. if number of distinct colors of current list at depth  $d$ , are lesser than target minus  $d$ , the sub tree is pruned.

**KEYWORDS:** Maximum clique, Coloring of graph, Heuristics, Color class.

## **I. INTRODUCTION**

Given a graph  $G(V, E)$ , where  $V$  is the set of vertices and  $E$  is the set of edges,  $d(v)$  is degree of vertex  $v \in V$ ,  $N(v)$  is a set of neighbors of vertex  $v$  and 'size' refers current size of sub graph. A complete sub graph of  $G$  is one whose vertices are pair wise adjacent. A clique is a complete sub graph  $S$  of a given graph  $G$ , where all vertices presented in  $S$  are mutually adjacent to each other. Finding a clique of a fixed size  $k$  is a well known NP-complete problem known as  $k$ -clique [8]. The corresponding optimization problem i.e. finding the maximum complete sub graph of  $G$  is known as the Maximum Clique Problem (MCP) [9]. MCP finds applications in many fields; coding theory, fault diagnosis, pattern recognition, and computational biology, Bioinformatics [10,11,12,13], etc. Being NP-complete problem, it is desired to develop an efficient maximum-clique finding algorithms that run very fast in practice. Since it is NP-Hard, it is difficult to obtain a polynomial time algorithm to find exact solution efficiently [9], but using various heuristics, the performance of algorithm may be obtained considerable. An independent set is a set of vertices whose elements are pair wise nonadjacent. The maximal independent set can be realized using greedy approach of graph coloring. Algorithms for the maximum clique problem have been proposed by many authors as in [1, 2, 3, 4,6,7]. They all are based on branch-and- bound, and backtracking techniques. They explore all the vertices sequentially with some pruning conditions. Unlike other branch and bound technique of MCP, the proposed algorithm (FCBB: fast color bases branch & bound) selects only a promising set of vertices for further expansion, i.e. a small set of vertices whose color is very close to or greater than & equal to target minus  $d$ , at depth  $d$ .

The Computational results with various DIMACS benchmark graphs show that the algorithm outperforms all the existing algorithm based on branch and bound techniques. The reason of better performance is perhaps the reduced search space at every depth, as at every recursive step only non adjacent vertices are likely to be further explored.

The rest of the paper is presented as follows, Section 2 discusses about related work. The proposed algorithm is discussed in Section 3. Comparisons of computational result for various DIMACS benchmark graphs are discussed in Section 4. The paper is concluded in Section 5.



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

## II. RELATED WORKS

Several papers are published since 1975s on MCP problem. Few of the most significant papers are referred here. The easiest and exact one was presented by Carraghan and Pardalos [1]. The algorithm uses a partial enumeration of vertices. It is generally used as a basis for computational comparison for any state-of-art algorithms. Another algorithm was published by P. Östergård [2] used a reverse ordering of vertices and asserted that the ordering of vertices have effect on computational time for different types of graph. He also has compared his algorithm with earlier published algorithms and has shown his

algorithm works better. P. Östergård algorithm, and is just a modifications of Carraghan and Pardalos' algorithm. One of the recent algorithm [3] presented by Deniss Kumlander, based on a fact that vertices from the same independent set couldn't be included into the same maximum clique. Those independent sets are obtained from a heuristic of vertex coloring where each of the independent set is a color class. The approach discussed in [6,7] uses a branch and bound to begins with a small clique and continues finding larger and larger cliques until one is found that can be verified to have the maximum size. They use a greedy approximate coloring in pruning condition, but in comparison with [3], it is far less efficient.

## III. PROPOSED ALGORITHM

The algorithm proposed (FCBB) is based on a heuristic that at each recursive step only a set vertices (one or more color classes) are explored further to get adjacent list (from the current list). The reason is that at the depth  $d$ , the maximum clique most likely would include one of the vertices of color class whose color is greater than or equal to target minus depth, as we know that chromatic number of a graph is greater than or equal to clique number of the graph  $G$ , i.e.  $\chi(G) \geq \omega(G)$ .

Unlike the algorithm in [3], here the coloring is done at each recursive step, as the coloring of residual graph (induced graph of current list of vertices) is more promising than the coloring of graph as a whole at the beginning. Coloring of graph is done using Dsatur algorithm given in [14].

Having a notion of depth, let vertex  $v_i$  at depth  $d$ , it considers all vertices at depth  $d$ , for finding adjacent to  $v_i$  and that becomes the current list at depth  $d+1$ , and so on. Let  $v_i$  is expanding currently at depth  $d$ , if the degree (number of distinct color) of the current list plus  $d <$  size of best clique (CBC) found so far, then it is pruned. The proposed algorithm is given in Fig. 1 below.

```
function clique(U, depth)
1. if |U|=0 then
2. if depth>max then
3. max =depth
4. return
5. V=Coloring_Find_Vset(U)
   // let k=target-depth, V ∈ U such that
   // k < color(v) < k+2, for every v ∈ V.
6. if(depth+degree(V)<=max)
7. return
   //degree (V) is number of distinct color in V.
8. if(max>=Target)
9. return
10. for i in 1 to |V|
11. v= U ∩ N(Vi) //v is adjacent vertices of Vi
12. clique(vi, depth + 1)
13. return max
.....
14. function main
15. max:=0
16. clique(V, 1)
17. return
```

Fig. 1: Proposed Algorithm.

# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Returning to depth 0, the CBC (max) is updated. It is observed that the vertices  $v \in U$  to be explored at depth  $d$ , are promising enough if  $k < \text{color}(v) < k+2$ , where  $k = \text{target}-d$ , and gives satisfactory result as shown in the Table-1.

## IV. EXAMPLE

The algorithm is described by taking an example shown in Figure-2, below.

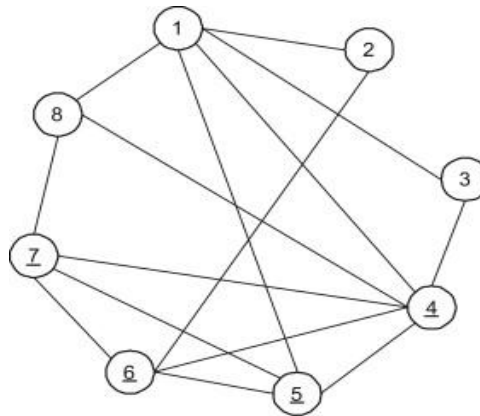


Fig. 2: A simple graph with clique size 4

Initial list of vertices  $U = \{1\ 2\ 3\ 4\ 5\ 6\ 7\ 8\}$

Here it can be shown that, very few vertices are qualified to be explored further.

**Depth 1:**  $U = \{1\ 2\ 3\ 4\ 5\ 6\}$ ,  $\text{max}=0, \text{target}=4, \text{depth}=1$ .

|        |   |   |   |   |   |   |   |   |
|--------|---|---|---|---|---|---|---|---|
| Vertex | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
| Color  | 2 | 1 | 3 | 1 | 3 | 2 | 4 | 3 |

select  $v = \{7\}$  //as  $\text{color}(7) > \text{target}-\text{depth}$ , i.e.  $4 > 3$

$N(7) = \{4\ 5\ 6\ 8\}$  //N:Neighbors

**Depth 2:**  $U = \{4\ 5\ 6\ 8\}$ ,  $\text{max}=0, \text{target}=4, \text{depth}=2$ .

|        |   |   |   |   |
|--------|---|---|---|---|
| Vertex | 4 | 5 | 6 | 8 |
| Color  | 1 | 2 | 3 | 2 |

Select  $v = \{6\}$  //as  $\text{color}(6) > \text{target}-\text{depth}$ , i.e.  $3 > 2$

$N(6) = \{4, 5\}$

**Depth 3:**  $U = \{4, 5\}$ ,  $\text{max}=0, \text{target}=4, \text{depth}=3$ .

|            |   |   |
|------------|---|---|
| Verte<br>x | 4 | 5 |
| Color      | 1 | 2 |



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

Select  $v = \{5\}$ , //as  $\text{color}(5) > \text{target-depth}$ , i.e.  $2 > 1$

$N(5) = \{4\}$

**Depth 4:**  $U = \{4\}$ ,  $\text{max} = 0, \text{target} = 4, \text{depth} = 4$

$V = \{4\}$

$N(4) = \Phi$

**Depth 5:**  $U = \{\Phi\}$

Depth  $>$  max,  $\text{max} = 4$

Return

Output:  $\text{max}(\text{Clique}) = 4 \{7, 6, 5, 4\}$

## V. COMPUTATIONAL RESULT

A comparative computational performance is enlisted in Table-1. The execution time is taken in seconds, the value  $\epsilon$  is mentioned when execution time is less than 1 millisecond, and value h is mentioned when execution time is more than  $\frac{1}{2}$  hour. They are executed on system with configuration; Intel(R) 3120M CPU -i3 @2.5 GHz. It is observed that the proposed algorithm (FCBB) outperforms the former three state-of-art algorithms presented in [1][2][3].

The Table-1 column 5, 6, 7 are execution time obtained by [1] (by R. Carraghan, P.M. Pardalos), [2] (Östergård), and [3] (by Deniss Kumlander) respectively as mentioned in reference. The eighth column is from the proposed one (FCBB).

TABLE 1: The execution time (in micro seconds) for various benchmark graphs

| Benchmark Graph  | #Vertices | Max Clique | Edge Density | [1]        | [2]        | [3]        | FCBB       |
|------------------|-----------|------------|--------------|------------|------------|------------|------------|
|                  |           |            |              | time(Sec)  | time(Sec)  | time(Sec)  | time(Sec)  |
| brock200_1.b     | 200       | 21         | 0.75         | 34.548     | 28.98      | 10.11      | 0.531      |
| brock200_2.b     | 200       | 12         | 0.5          | 0.078      | 0.015      | 0.062      | 0.015      |
| brock200_3.b     | 200       | 15         | 0.61         | 0.578      | 0.437      | 0.546      | 0.09       |
| brock200_4.b     | 200       | 17         | 0.66         | 2.906      | 1.64       | 1          | 0.188      |
| brock400_1.b     | 400       | 27         | 0.75         | h          | h          | h          | 20.67      |
| brock400_2.b     | 400       | 29         | 0.75         | h          | h          | h          | 21         |
| brock400_3.b     | 400       | 31         | 0.75         | h          | h          | h          | 65         |
| brock400_4.b     | 400       | 33         | 0.75         | h          | h          | h          | 9.5        |
| c-fat200-1.clq   | 200       | 12         | 0.16         | $\epsilon$ | $\epsilon$ | $\epsilon$ | $\epsilon$ |
| c-fat200-2.clq   | 200       | 24         | 0.16         | $\epsilon$ | $\epsilon$ | $\epsilon$ | $\epsilon$ |
| c-fat200-5.clq   | 200       | 58         | 0.43         | 0.046      | $\epsilon$ | $\epsilon$ | $\epsilon$ |
| c-fat500-1.clq   | 500       | 14         | 0.04         | $\epsilon$ | $\epsilon$ | $\epsilon$ | $\epsilon$ |
| c-fat500-10.clq  | 500       | 126        | 126          | 0.016      | 0.017      | $\epsilon$ | 0.016      |
| c-fat500-2.clq   | 500       | 26         | 0.37         | $\epsilon$ | $\epsilon$ | $\epsilon$ | $\epsilon$ |
| c-fat500-5.clq   | 500       | 64         | 0.07         | 0.156      | $\epsilon$ | $\epsilon$ | $\epsilon$ |
| gen200_p0.9_44.b | 200       | 44         | 0.9          | h          | h          | h          | 0.031      |



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

|                         |                  |                   |                     |            |            |            |             |
|-------------------------|------------------|-------------------|---------------------|------------|------------|------------|-------------|
| gen200_p0.9_55.b        | 200              | 55                | 0.9                 | h          | h          | 34.532     | 0.015       |
| hamming6-2.clq          | 64               | 32                | 0.9                 | €          | €          | €          | €           |
| hamming6-4.clq          | 64               | 4                 | 0.35                | €          | €          | €          | €           |
| hamming8-2.clq          | 256              | 128               | 0.97                | €          | 0.016      | €          | 0.032       |
| hamming8-4.clq          | 256              | 16                | 0.64                | €          | €          | €          | €           |
| <b>Benchmark Graphs</b> | <b>#Vertices</b> | <b>Max Clique</b> | <b>Edge Density</b> | <b>[1]</b> | <b>[2]</b> | <b>[3]</b> | <b>FCBB</b> |
|                         |                  |                   |                     | time(Sec)  | time(Sec)  | time(Sec)  | time(Sec)   |
| johnson8-4-4.clq        | 70               | 14                | 0.77                | €          | €          | €          | €           |
| johnson16-2-4.clq       | 120              | 8                 | 0.76                | €          | €          | €          | €           |
| johnson32-2-4.clq       | 496              | 16                | 0.76                | €          | 0.078      | 0.015      | 0.016       |
| keller4.clq             | 171              | 11                | 0.65                | €          | 0.015      | €          | 0.016       |
| keller5.clq             | 776              | 27                | h                   | h          | h          | h          | 12.34       |
| MANN_a9.clq.b           | 45               | 16                | 0.99                | €          | €          | €          | €           |
| MANN_a27.clq            | 378              | 126               | 0.93                | h          | h          | 45.14      | 0.047       |
| MANN_a45.clq            | 1035             | 345               | h                   | h          | h          | h          | 21.98       |
| p_hat300_1.clq          | 300              | 8                 | 0.24                | 0          | 0.062      | 0.015      | 0.015       |
| p_hat300_2.clq          | 300              | 25                | 0.49                | 1.43       | 4.328      | 0.141      | €           |
| p_hat300_3.clq          | 300              | 36                | 0.74                | h          | h          | h          | 0.203       |
| p_hat500-1.clq          | 500              | 9                 | 0.25                | 0          | 0          | 0.156      | 0.015       |
| p_hat500-2.clq          | 500              | 36                | 0.5                 | h          | h          | h          | €           |
| p_hat500-3.clq          | 500              | 50                | 0.75                | h          | h          | h          | 105.86      |
| p_hat700-1.clq          | 700              | 11                | 0.748               | 0.406      | 0.188      | 0.672      | 0.063       |
| p_hat700-2.clq          | 700              | 44                | 0.498               | h          | h          | h          | 0.125       |
| p_hat700-3.clq          | 700              | 62                | 0.748               | h          | h          | h          | 31.18       |
| p_hat1000-1.clq         | 1000             | 10                | 0.245               | 0.047      | 0.109      | 0.031      | 0.016       |
| p_hat1000-2.clq         | 1000             | 46                | 0.49                | h          | h          | h          | 0.032       |
| p_hat1500-1.clq         | 1500             | 12                | 0.253               | 31.86      | 2.79       | 11.36      | 0.39        |
| p_hat1500-2.clq         | 1500             | 65                | 0.506               | h          | h          | h          | 0.078       |
| san200_0.7_1.clq        | 200              | 30                | 0.7                 | h          | h          | h          | €           |
| san200_0.7_2.clq        | 200              | 18                | 0.7                 | h          | h          | 0.015      | 0.015       |
| san200_0.9_1.clq        | 200              | 70                | 0.9                 | h          | h          | 17.92      | €           |
| san200_0.9_2.clq        | 200              | 60                | 0.9                 | h          | h          | 27.438999  | 0.016       |
| san200_0.9_3.clq        | 200              | 44                | 0.9                 | h          | h          | h          | 0.406       |
| san400_0.5_1.clq        | 400              | 13                | 0.5                 | h          | h          | 0.094      | 0.016       |
| san400_0.7_2.clq        | 400              | 30                | 0.7                 | h          | h          | 3.594      | 0.156       |
| san400_0.7_3.clq        | 400              | 22                | 0.7                 | h          | h          | 3.71       | 1.141       |
| san400_0.9_1.clq        | 400              | 100               | 0.9                 | h          | h          | h          | 0.75        |
| sanr200_0.7.clq         | 200              | 18                | 0.7                 | 5.04       | 1.75       | 6.485      | 0.125       |



# International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 6, June 2016

|                 |     |    |       |       |      |      |        |
|-----------------|-----|----|-------|-------|------|------|--------|
| sanr200_0.9.clq | 200 | 42 | 0.9   | h     | h    | h    | 18.024 |
| sanr400_0.5.clq | 400 | 13 | 0.5   | 0.344 | 1.59 | 5.53 | 0.11   |
| sanr400_0.7.clq | 400 | 21 | 0.7   | h     | h    | h    | 0.445  |
| C125.9.clq      | 125 | 34 | 0.898 | h     | h    | 5.31 | 0.218  |

## VI. CONCLUSION

This paper proposes a color based efficient heuristic approach for finding maximum clique taking advantage of color class or maximal independent set of vertices having color very close to the remaining length of clique (target) at depth  $d$ . The algorithm is efficient because the search space gets drastically decreased, as at each recursive step. Some benchmark graphs like brock 800\*.cp, P\_hat1000.3.cq, P\_hat1500.3.cq are appeared to be difficult to compute, for remaining almost all of the benchmark graphs, the algorithm gives satisfactory result. The experimental results show that the vertices  $v \in U$  at depth  $d$ , to be further expanded, are promising enough if  $k < \text{color}(v) < k+2$ , where  $k = \text{target} - d$ , and gives satisfactory result as shown in the Table-1.

## REFERENCES

- [1] R. Carraghan, P.M. Pardalos, "An exact algorithm for the maximum clique problem", Oper. Research. Letters. Vol. 9 , pp 375–382, 1990.
- [2] Östergård, P.R.J., "A fast algorithm for the maximum clique problem". Volume 120, Issues 1–3, pp 197–207, 15 August 2002.
- [3] Deniss Kumlander, "A simple and efficient algorithm for the maximum clique finding reusing a Heuristic vertex colouring". IADIS international journal on computer science and information systems, Vol. 1, No. 2, pp. 32-49, 2006.
- [4] D. Kumlander, "A new exact algorithm for the maximum-weight clique problem based on a heuristic vertex-coloring and a backtrack search", Proceedings of The Forth International Conference on Engineering Computational Technology, Civil-Comp Press, , pp. 137-138, 2004.
- [5] Ashay Dharwadker, "The Maximum Clique problem", published by Amazon, 2011.
- [6] Etsuji Tomita, Tatsuya Akutsu and Tsutomu Matsunaga: "Efficient Algorithms for Finding Maximum and Maximal Cliques: Effective Tools for Bioinformatics", ISBN 978-953-307-475-7, 2011.
- [7] Tomita, E., Sutani, Y., Higashi, T., Takahashi, S., Wakatsuki, M.: "A simple and faster branch-and-bound algorithm for finding a maximum clique", WALCOM: Algorithms and Complexity, Lecture Notes in Computer Science, 5942, pp. 191-203, 2010
- [8] Karp, R.M: In: Miller, R.E., Thatcher, J.W. (eds.) "Reducibility among Combinatorial Problems", Plenum, pp. 85–103, 1972.
- [9] Bomze, I.M., Budinich, M., Pardalos, P.M., Pelillo, M.: "HandBook of Combinatorial Optimization. Supplement A". Kluwer Academic Publishers, pp. 1–74, 1999.
- [10] Butenko, S., Wilhelm, W.E.: "Clique-detection models in computational biochemistry and genomics".Eur.J.Operat.Res.173, pp. 1–17, 2006.
- [11] C. Hofbauer, H. Lohninger, A. Aszodi, SURFCOMP: A novel graph based approach to molecular surface comparison, Journal of Chemical Information and Computer Science 44, pp. 837-847, 2004.
- [12] S. Schmitt, D. Kuhn, G. Klebe, A new method to detect related function among proteins independent of sequence and fold homology, Journal of Molecular Biology 323 pp. 387-406, 2002.
- [13] J. Konc, D. Janežič, A branch and bound algorithm for matching protein structures, Lecture Notes in Computer Science Vol. 4432, pp. 399-406, 2007.
- [14] Matula, D. Marble, G., and Isaacson, J. Graph coloring algorithms. In Graph Theory and Computing, Academic Press, New York, , pp. 109-122, 1972.