



# Alternative Branching Strategies in the Branch and Bound Algorithm by Using a K-Clique Covering Vertex Set for Maximum Clique Problems

Mochamad Suyudi<sup>1</sup>, Asep K. Supriatna<sup>2</sup>, Sukono<sup>3</sup>

<sup>1,2,3</sup>*Department of Mathematics, Universitas Padjadjaran, Indonesia*

*\*Corresponding author email: moch.suyudi@gmail.com*

---

## Abstract

The maximum clique problem (MCP) is graph theory problem that demand complete subgraph with maximum cardinality (maximum clique) in arbitrary graph. Solving MCP usually use Branch and Bound (BnB) algorithm. In this paper, we will show how  $n + 1$  color classes (where  $n$  is the difference between upper and lower bound) selected to form k-clique covering vertex set which later used for branching strategy can guarantee finding maximum clique.

*Keywords:* maximum clique problem, graph theory problem, bound algorithm

---

## 1. Introduction

The maximum clique problem (MCP) is the problem of finding a complete subgraph with the maximum number of vertices in it (maximum clique) on any graph  $G = (V, E)$ , the maximum clique cardinality in  $G$  is denoted by  $\omega(G)$ . The maximum clique has a wide range of applications including social network analysis (Balasundaram et al., 2011), economics (Boginski et al., 2006), bioinformatics (Szabó and Zavalnij, 2018), and biochemistry (Pardalos and Xue, 1994). In solving MCP generally using the Branch and Bound (BnB) algorithm (Wu et al., 2019), examples of the BnB algorithm for MCP can be found in (Aslan and Baykan, 2016) and a review of the algorithm for MCP can be seen in (Wu et al., 2019; Blum et al., 2020; Smith et al., 2019; Guo et al., 2019). Pardalos and Xue (1994) formulated three main keys to the BnB algorithm for MCP, namely:

- (1) How is a good lower bound?
- (2) How is a good upper bound of the maximum clique size determined?
- (3) How is the search carried out?

According to Babel and Tinhofer, G. (1990), there are two stages in the BnB algorithm, namely the bounding stage and the branching stage, the branching stage is carried out to break the problem into smaller sub-problems (in the form of subgraphs) so that the search for maximum clique will be easier to do, then at the limitation stage, the upper bound (UB) and the lower bound (LB) are determined to cancel the subgraph of the branching result which will not produce maximum clique. Szabó and Zavalnij (2019) describe how a  $k$ -clique covering vertex set is used for branching on BnB for MCP. In this paper, the method for determining the  $k$ -clique covering vertex set uses a color class, but they note that this method can guarantee that the set has a minimum size so that it will affect the length of the branching process, then we realize that choosing an incorrect color class can causing no maximum clique to be found. So that it is motivated by this in this paper to explain how the selection of color classes is carried out to form the  $k$ -clique covering vertex set which will later be developed into a branching strategy in BnB for MCP.

## 2. Definitions and Notation

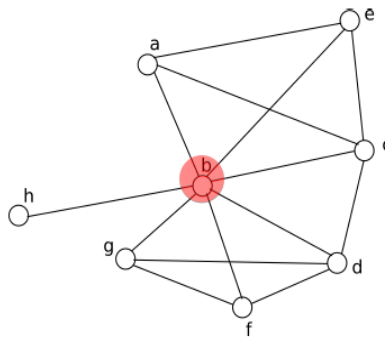
In this paper, the graph used is a simple graph, the graph  $G = (V, E)$  is said to be a simple graph if the graph does not have directed edges, double edges and loops. The set of vertices which are adjacent to vertex  $v$  is denoted by  $N(v)$ , while the neighborhood set of  $v$  is  $N[v] = N(v) \cup v$ . Let  $G = (V, E)$  be a simple graph and  $X \subseteq V$ , the subgraph  $H$  resulting from subtracting the set  $V - X = Y$  on graph  $G$  is called an induced subgraph denoted by  $H[Y]$ .

Clique is a subgraph in which every vertex within is adjacent to one another,  $k$ -clique is clique with  $k$  vertices (Yang et al., 2019; Blanuša et al., 2020).  $k$ -clique is said to be maximum on the graph if there is no  $(k + 1)$ -clique on the graph, the maximum cardinality of clique in  $G$  is called the clique number  $G$  denoted by  $\omega(G)$ . Graph coloring is the color labeling of vertices with the condition that there are no the same color on two adjacent vertices, a set of vertices that have the same color is called a color class, in this paper the color class will be denoted by  $CC_n$  where the number of  $n$  is as many as the colors on the graph. The minimum number of colors from the coloring results on graph  $G$  is called the chromatic number denoted by  $\chi(G)$ . Let  $C$  be the maximum clique on  $G$ , the coloring of the graph against  $C$  requires exactly as much color  $\chi(G)$  and since  $C$  is a subgraph of  $G$ , then the coloring on  $G$  is at least as much  $\omega(G)$  or

$$\omega(G) \leq \chi(G) \tag{1}$$

Let  $G = (V, E)$  is a simple finite graph and  $W \subseteq V$ , if each  $k$ -clique has at least one vertex on  $W$ , then  $W$  is called the  $k$ -clique covering vertex set (see Szabó and Zavalnij, 2018). An illustration of the  $k$ -clique covering vertex set can be seen in Figure 1.

The 4-clique covering vertex set on the graph in Figure 1 contains vertex  $b$ , but if selected the set of vertices  $c, d$ , then the 4-clique covering vertex set is a set of vertices covering  $k$ -clique on a graph is not a unique set.



**Figure 1.** 4-clique covering vertex set.

### 3. Branching with vertices covering the k-clique

This section describes the results of study and discussions which are taken from transcription of interview and questionnaire. The findings are intended to answer the purposes of study which include analyzing how the teacher integrated technologies in blended learning, how the students perceive blended learning using technologies, and what challenges overcome by the teacher and students during blended learning.

Let  $W = \{v_1, v_2, \dots, v_n\}$  is a  $k$ -clique covering vertex set, and let  $\Delta$  is  $k$ -clique in graph  $G$ . If an induced subgraph  $H_i = G[N[v_i]]$  is formed where  $1 \leq i \leq n$  according to the definition of the set of vertices covering  $k$ -clique, then in the induced subgraph  $k$  vertices of  $v$  or in other words there are  $k$ -clique in  $H_i$ . By making induced subgraphs all  $k$ -clique have been mapped in each subgraph so that it is easier for us to find  $k$ -clique on graph  $G$ .

To find  $k$ -clique covering vertex set where  $k$  is the maximum can be obtained from the vertex coloring of the graph. Let  $G = (V, E)$  is a simple graph that has been colored vertices on it where  $\omega(G) = k$ , it will be clear that the set of vertices that make up the maximum clique has a different color on each vertex by as many as  $k$  colors. Let  $\Delta$  is the maximum clique in  $G$  and  $C_n$  is the color class, in this condition if an induced subgraph is formed from  $v \in C_n \cap \Delta$  this will result in the induced subgraph, or in other words we only need to select the color class where the color is is on that which has been colored. However, it is difficult to know whether the colors in the selected color class are included in the maximum clique color.

**Example:** will illustrate the process of branching by utilizing the  $k$ -clique covering vertex set. Notice that the graph  $G$  below has  $\chi(G) = 4$  and  $\omega(G) = 4$  where the color classes are  $C_1 = \{a, d, g, 0\}$ ,  $C_2 = \{b, c, m\}$ ,  $C_3 = \{e, h, l, n\}$ , and  $C_4 = \{f, i, j, k\}$ . For a branch on  $G$ , let the  $k$ -clique covering vertex set from  $C_2$  is chosen, so that the branching will produce three new subgraphs, namely  $H_1 = G[N[b]]$ ,  $H_2 = G[N[c]]$ , and  $H_3 = G[N[m]]$ . Subgraph  $H_1$  can be seen easily as 4-clique as well as the maximum clique in  $G$ , namely  $\{a, b, h, i\}$  and  $\{b, g, k, n\}$ , then in  $H_2$  there is also a 4-clique namely  $\{a, c, e, j\}$ , and  $\{c, d, f, l\}$ .

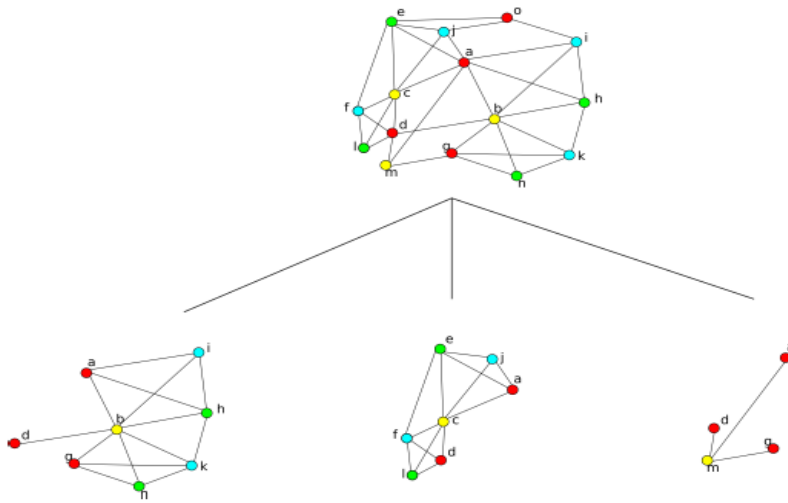


Figure 2. The 4-clique covering vertex set.

It should be noted, the method of finding a  $k$ -clique covering vertex set with a color class does not guarantee a minimum set size, meaning that the more vertices in the color class, the more subgraphs will be formed so that it will take more time to work on it. On the other hand, not all vertices in the color class can produce maximum clique in its branching subgraph as subgraph  $G = [N [m]]$  above. The use of a color class where the color is not used in the maximum clique coloring causes no maximum clique in the branching subgraphs, see the Figure 3.

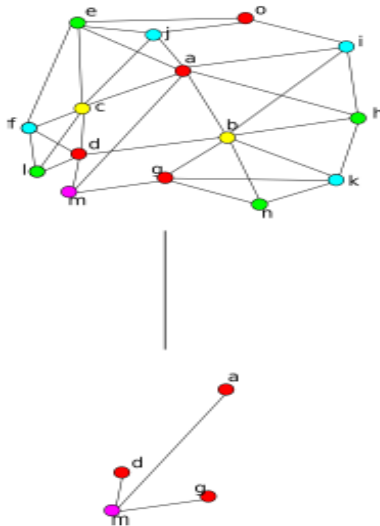


Figure 3. Branching with a purple color class

In Figure 3, the use of the purple color class for the branches does not produce a maximum clique in it, this is because purple is not included in the maximum clique color combination (red, yellow, green, blue).

### 3.1. Selection of color classes.

According to equation (1), for every graph  $\omega(G) \leq \chi(G)$ , so that the branching stage of the search for clique on the graph will be viewed according to two conditions, namely the graph with  $\omega(G) = \chi(G)$  and  $\omega(G) < \chi(G)$ . On the graph  $\omega(G) = \chi(G)$ , the branching process of finding a clique is easier than  $\omega(G) < \chi(G)$ , because a graph with  $\omega(G) = \chi(G)$  has only one combination of clique coloring the maximum is  $C(\chi(G), \omega(G)) = 1$ , meaning that all the colors on the graph are used for maximum clique coloring so that only taking one color class can be ensured that the set of vertices covers the  $k$ -clique  $W \subseteq C_n$  for every  $1 \leq n \leq \chi(G)$ . However, if a graph has  $\omega(G) < \chi(G)$ , then this will result in more than one color combination is  $C(\chi(G), \omega(G)) = \frac{\chi(G)!}{\omega(G)!(\chi(G)-\omega(G))!} > 1$ , because it is not known how many clique the maximum number of clique is in graph  $G$  and the form of graph, that way the graph has the most possible color combinations in the maximum clique  $\frac{\chi(G)!}{\omega(G)!(\chi(G)-\omega(G))!}$  and there is at least one maximum clique color combination on the graph. Inappropriate selection of color classes can result in not all of the maximum clique found by the clique search branching subgraph until the worst case, namely the absence of maximum clique in these subgraphs. In the condition  $\omega(G) < \chi(G)$ , it is difficult to determine how many color combinations are used to form the maximum clique on the colored graph.

**Theorem 3.2.** Let  $n, r \in \mathbb{Z}^+$  where  $1 \leq r \leq n$ , then;

$$C(n, r) = C(n-1, r-1) + C(n-2, r-1) + \dots + C(r-1, r-1).$$

**Proof.** The subset combination will be divided according to the smallest to largest number sequentially. Suppose the combination starts from number 1, the number of combinations based on number 1 will be as much as  $C(n-1, r-1)$ . Then the number of the next combination based on number 2 will be as much as  $C(n-2, r-1)$ ,  $n-2$  because number 1 has been loaded in the previous combination. So that the combination with the number  $x$  is as much as  $C(n-x, r-1)$ , the subset group will stop at number  $(y = n-r) + 1$  where  $C(n-(y+1), r-1) = C((n-y)-1, r-1) = C(r-1, r-1) = 1$ . Combines all combinations based on numbers from 1 until  $y+1$  gives  $C(n, r)$ , so **Theorem 3.2** is proven.

In the previous explanation, the use of color classes where the color is in the maximum clique color combination can find the maximum clique in the subgraph of the results of the branching, to overcome the problem of not knowing which maximum clique color combination is present in a graph, in this paper it will be assumed that there are all color combinations maximum clique on the graph. To reduce the number and size of subgraphs resulting from the branching, color classes with cardinality and quantity will be selected minimum degree. Suppose on any graph  $G$  has vertices with degrees  $|V| - 1$ , if  $G$  is colored by vertices on them, vertices with degrees  $|V| - 1$  will become color classes with the vertices themselves. The induced subgraph of the vertex of the color class will produce the same subgraph as the previous graph so that the color class to be avoided in the color

class selection is the vertex having the degree  $|V| - 1$ . In the selection of more than one color class, the formation of induced subgraphs with vertices in the color class must be reduced by the color class that has been used, for example, the  $CC_1, CC_2, CC_3$  color classes are selected, then the induced subgraph formation with these color classes is sequentially  $(CC_3, CC_1, CC_2)$  then all  $CC_1$  induced subgraphs will be reduced by  $CC_3$  then all  $CC_2$  induced subgraphs will be reduced  $CC_3 \cup CC_1$ .

#### 4. BnB Algorithm

According to Wu and Hao (2015) to solve MCP generally using BnB. Babel and Tinhofer, (1990) states there are two stages in the BnB algorithm, namely the bound stage and the branching stage. The previous explanation has discussed the use of a set of vertices covering the  $k$ -clique as a branching strategy and how to select color classes to be used as a set. This method will be used for the branching stage of this paper, then for the bound stage an algorithm is needed to determine the upper and lower bound. Algorithms for determining the upper bound and lower bound can be found in (Babel and Tinhofer, 1990; Suyudi et al., 204), various comparisons of the algorithm for determining the upper bound can be seen in (Aslan and Baykan, 2016). For each clique on graph  $G$  is the lower bound (LB) of  $\omega(G)$  and for each color on the graph  $G$  is the upper bound (UB) of  $\chi(G)$  (Suyudi et al., 204),:

$$LB \leq \omega(G) \leq \chi(G) \leq UB \quad (2)$$

Suppose that  $UB^*$  is the upper bound on subgraph  $H$ , if  $UB^* \leq LB$  then according equation (2)  $\omega(H) \leq LB$  or subgraph  $H$  will not have a maximum clique greater than LB. On this basis, the search for maximum clique will only focus on subgraphs with  $UB^* \geq LB$  and pruning subgraphs that do not meet these requirements. The following is an example of an algorithm for determining the lower bound (Babel and Tinhofer, 1990; Suyudi et al., 204):

---

#### Algorithm 1: Maximum Clique

---

```

Let  $V := \{v_1, v_2, \dots, v_n\}$  is set vertices of  $G$ 
 $W := V$ 
CLIQUE :=  $\emptyset$ 
While  $W \neq \emptyset$  do
    select any  $v \in W$  where  $\deg(v) = \Delta G$ 
    CLIQUE  $\cap \{v\}$ 
     $W := W \cap N(v)$ 

```

---

In determining the upper limit, we can use node coloring algorithms such as DSATUR, Welsh-Powell (WP), Incidence Degree Ordering, etc. This paper will show an example of the WP algorithm as follows:

**Algorithm 2: Welsh-Powell**


---

```

Let  $V := \{v_1, v_2, \dots, v_n\}$  is set vertices of  $G$ 
 $W := V$ 
COLOR := 0
select  $v \in W$  where  $\deg(v) = \Delta G$ 
ColorClass(COLOR) :=  $\emptyset$ 
ColorClass(COLOR)  $\cup \{v\}$ 
 $W - \{v\}$ 
while  $W \neq \emptyset$  do
  for 1 to  $|W|$  do
     $X := W$ 
    select  $v \in X$  where  $\deg(v) = \max(N(v))$ 
    if  $v$  not adjacent to  $w \in \text{ColorClass}(\text{COLOR})$  then
      put  $v$  in ColorClass(COLOR)
       $X - \{v\}$ ;
    else
       $X - \{v\}$ 

```

---

The BnB algorithm in this paper begins with the initial value of  $\omega(G)$ ,  $k = 0$ , the value of  $k$  is used to select all graphs or subgraphs in the  $k$ -level tree search for upper and lower bound searches and branches, then the  $k$  value will continue to increase as iteration progresses. Furthermore, after searching the upper bound of all graphs at  $k$ -level, the lower bound of the graph with  $UB > \omega(G)$  will be determined, graphs that do not meet these conditions will be pruned. The value of  $\omega(G)$  will continue to increase following the found LB if  $LB > \omega(G)$  then the CLIQUE set obtained in Algorithm 1. Maximum Clique will be said to be maximum if  $UB = LB$ . Determination of upper and lower bound will stop and continue to the branching stage if all graphs at level- $k$  have been reviewed. In the branching stage, it is carried out by following the conditions for selecting the color class as explained earlier, the graph will be branched if  $UB \geq \omega(G)$  if it does not meet these conditions then the graph will be pruned. The following is the overall BnB algorithm:

**Algorithm 3: Branch and Bound**


---

```

Step (0) Initialization
 $\omega(G) := 0, k := 0; \text{CLIQUE} := \emptyset$ 
Step (1) Graph selection
select all graphs at level- $k$ 
 $k := k + +$ 
go to (2)
Step (2) Define upper and lower bound
do
  Find UB on the graph
  if  $UB \geq \omega(G)$  then
    find LB

```

---

---

```

if  $\omega(G) < LB$  then
   $\omega(G) := LB$ 
if  $UB = LB$  and  $i > |W(G)|$  then
  CLIQUE is the maximum clique
else
  Prune
while all graphs at level- $k$  have been defined  $UB$  and  $LB$ ;
if there are no more branchable graphs then
  search stopped
else
  go to (3)
Step (3) Branching do
  if  $UB \geq \omega(G)$  then
  graph branch
  else
  Prune
while all graphs at level- $k$  have been branched;
go to (1)

```

---

## 5. Conclusion

It has been explained how the use of color classes as a branching strategy in the BnB algorithm for MCP, in selecting the color class it is necessary  $(n = UB - LB) + 1$  color class in order to find the maximum clique. The overall BnB algorithm has also been shown in this paper but it is necessary to make a program for the algorithm in order to see the algorithm's performance.

## References

- Aslan, M., and Baykan, N. A. (2016). A performance comparison of graph coloring algorithms. *International Journal of Intelligent Systems and Applications in Engineering*, 1-7.
- Babel, L., and Tinhofer, G. (1990). A branch and bound algorithm for the maximum clique problem. *Zeitschrift für Operations Research*, 34(3), 207-217.
- Balasundaram, B., Butenko, S., and Hicks, I. V. (2011). Clique relaxations in social network analysis: The maximum k-plex problem. *Operations Research*, 59(1), 133-142.
- Blanuša, J., Stoica, R., Ienne, P., and Atasu, K. (2020). Manycore clique enumeration with fast set intersections. *Proceedings of the VLDB Endowment*, 13(12), 2676-2690.
- Blum, C., Djukanovic, M., Santini, A., Jiang, H., Li, C. M., Manyà, F., and Raidl, G. R. (2020). Solving longest common subsequence problems via a transformation to the maximum clique



- problem. *Computers & Operations Research*, 125, 105089.
- Boginski, V., Butenko, S., and Pardalos, P. M. (2006). Mining market data: a network approach. *Computers & Operations Research*, 33(11), 3171-3184.
- Guo, P., Wang, X., Zeng, Y., and Chen, H. (2019). MEAMCP: A Membrane Evolutionary Algorithm for Solving Maximum Clique Problem. *IEEE Access*, 7, 108360-108370.
- Pardalos, P. M., and Xue, J. (1994). The maximum clique problem. *Journal of global Optimization*, 4(3), 301-328.
- Smith, D. H., Perkins, S., and Montemanni, R. (2019). Solving the maximum clique problem with a hybrid algorithm. *International Journal of Metaheuristics*, 7(2), 152-175.
- Suyudi, M., Mohd, I. B., Mamat, M., Sopiyan, S., and Supriatna, A. K. (2014). Solution of maximum clique problem by using branch and bound method. *Appl. Math. Sci*, 8(2), 81-90.
- Szabó, S., and Zavalnij, B. (2018). Decomposing clique search problems into smaller instances based on node and edge colorings. *Discrete Applied Mathematics*, 242, 118-129.
- Szabó, S., and Zavalnij, B. (2019). Benchmark problems for exhaustive exact maximum clique search algorithms. *Informatica*, 43(2), 24-35.
- Wu, K., Gao, J., Chen, R., and Cui, X. (2019). Vertex Selection Heuristics in Branch-and-Bound Algorithms for the Maximum k-Plex Problem. *International Journal on Artificial Intelligence Tools*, 28(05), 1950015.
- Wu, Q., and Hao, J. K. (2015). A review on algorithms for maximum clique problems. *European Journal of Operational Research*, 242(3), 693-709.
- Yang, Y., Yan, D., Zhou, S., and Guo, G. (2019, November). Parallel clique-like subgraph counting and listing. In *International Conference on Conceptual Modeling* (pp. 484-497). Springer, Cham.