Applied Mathematical Sciences, Vol. 8, 2014, no. 2, 81 - 90
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2014.310601

Solution of Maximum Clique Problem by Using Branch and Bound Method

Mochamad Suyudi[1], Ismail Bin Mohd[2], Mustafa Mamat[3, 6], Sudrajat Sopiyan[4] and Asep K. Supriatna[5]

[1,4,5] Department of Mathematics,
University of Padjadjaran, Indonesia.
[2,3] Research Fellow, Department of Mathematics
Universiti Malaysia Terengganu, Malaysia
[6]Faculty of Informatics and Computing
Universiti Sultan Zainal Abidin, Terengganu, Malaysia

Abstract

The maximum clique problems have been widely applied in the field of operations research. The maximum clique problem is finding a solution that is often considered difficult, as well as graph coloring problem, traveling salesman, and so forth. Solution the maximum clique can be done using several methods, such as by using brute force, backtracking, and branch and bound. This paper discusses the branch and bound procedure to solve the maximum clique problem. In the discussion of the method described on the upper and lower bound by implementing a greedy heuristic and branching procedure for determining the maximum clique $\omega(G)$ and minimum coloration $\chi(G)$ in the graph G. Solution the maximum clique problem with this method is beneficial because it provides an alternative perspective on branch and bound, but it is also because they do not need to explain software optimization.

Mathematical Subject Classification: 90C59, 68R10, 05C15

Keywords: Maximum Clique, branch and bound, graph coloring, heuristic, greedy

1. Introduction

Solution the maximum clique problem in graph coloring using only the greedy algorithm would have difficulty [1]. Therefore, to facilitate maximum clique search here will use the branch and bound. Branch and bound is usually used as a method of resolving integer programming (IP), IP is a constraint integer from fractional (relaxation) and linear programming (LP) relaxation of the corresponding resolved to produce an upper bound or lower bound, see [9].

Solution the branch and bound method in the context of the maximum clique problem is considered easy and simple to execute, through the branch and bound procedure [2, 3]. Procedures branch and bound method is to determine the clique number and chromatic number of a graph. For a more traditional approach to branch and bound, can refer to [4, 5, 9].

A description of the basic concepts of graph theory to a wider survey refer to [6], [7] and [8]. Since the branch and bound on pure integer programming area can seem abstract, use simple graph algorithm to describe the procedure can be made more visual and easy to understand. Hence, in this paper examined on completion maximum klique using the branch and bound.

2. Graph Coloring

A graph $G = (V, E)$ is a set of vertices and edges. (Along this paper, the vertex will be represented by letters and colors for the vertices will be represented by integers). A clique in G is part of the vertices $C \subseteq V$ where each pair of vertices are joined with an edge, for example, see Figure 1. In other words, all the vertices in the clique must adjasen one another. The size of the maximum clique in G indicated by $\omega(G)$. An illustration of a nice, simple, this problem is with a dinner party. Suppose the vertices of the graph represent the dinner guests. Suppose edge connects guests if they already know each other. Then the maximum clique will be the biggest set of guests at a dinner where everyone knew each other [4, 5].

A vertices coloring is to assign a color to the vertices of a graph G such that no two adjacent vertices receive the same color. A minimum coloring is a coloring that uses the least amount possible for the entire graph coloring, as an example, see Figure 2. The size of the minimum coloring in G indicated by $\chi(G)$. Minimum Coloring can also be illustrated using the example of a dinner party described above. Suppose that the host wants her friends to meet each other. Then it will just sit people at tables where they are not familiar with the other guests. If we let the different colors represent the different tables at the party, then the only guests at no edge connecting them (ie, guests who do not know each other) will be allowed to color / same table. Coloring minimum in accordance with the minimum number of tables required to ensure no guests sit with people they've known [2, 4, 5].

Consider a triangle, which is a clique of size 3. Each of the three vertices must have different colors, if not, then there will be no edge has endpoints with the same color and coloring will be valid. In general, in any coloring a graph G, each pair of adjacent vertices must have different colors. Therefore, $\omega(G) \leq \chi(G)$.

More importantly, every clique in G is a lower bound on $\omega(G)$, and every shade in G is an upper bound on $\omega(G)$. In the next section, given a heuristic for determining the clique and color and, hence, obtain the boundaries to be used in a branch and bound algorithm [6; 9].

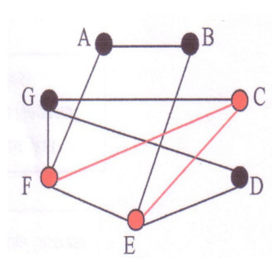


Figure 1. Vertices C, E, and F form clique size 3

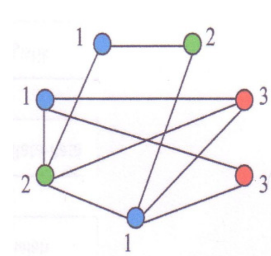


Figure 2. Shows the Coloring of Size 3

3. Procedures of Branch and Bound

3.1 Determine the Upper and Lower Bound

To find a lower bound (LB) on the size of the maximum clique, the search heuristic greedy clique can be implemented in a simple graph $G = (V, E)$ as follows:

1. Suppose $K = \emptyset$.
2. Suppose v is a vertices in G but not in K with the maximum degree. Adding v in K.
3. Remove all vertices that are not adjasen to v from V.
4. If V is empty, then stop. If not, then go back to step 2.

Basically, added to the clique K vertices with the greatest degree that is connected to all the vertices, and are already in the clique. The cardinality of the set K is generated is a clique in the graph G and therefore, lower bound on $\omega(G)$, is the size of the maximum clique in G [2, 7]. Figure 3 illustrates this heuristic. For more details, look at the following discussion.

Step 1. Let $K = \{\ \}$. Determine the maximum vertices degree, the vertices F (Figure 3.a.);

Step 2. Adding F to K, so that sehingga $K = \{F\}$, then remove C is not adjasen with F (Figure 3.b.);

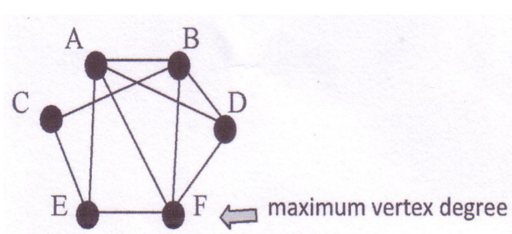


Figure 3.a.

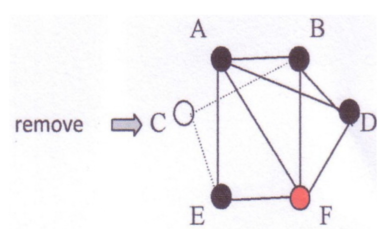


Figure 3.b.

Step 3. Adding D to K, so that $K = \{F, D\}$, which is not clear adjasen E with D (Figure 3.c.);

Step 4. Adding B to K, so that $K = \{F, D, B\}$(Figure 3.d.);

Step 5. Adding A to K, so that K = {F, D, B, A}. In order to get the maximum clique, ie, $\omega(G) \geq 4$ = lower bound (LB) (Figure 3.e.).

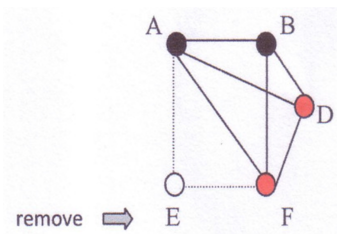


Figure 3.c.

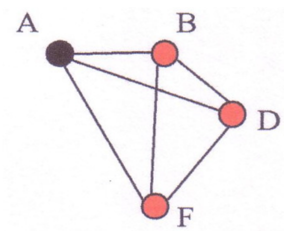


Figure 3.d.

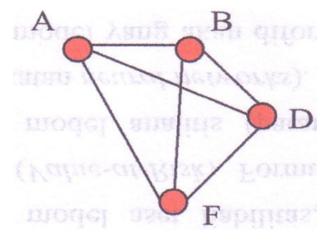


Figure 3.e.

Figure 3. a, b, c, d, and e, is an illustration of the heuristic greedy clique, which $\omega(G) \geq 4$.

To find the upper bound (UB) on the size of the maximum clique, (heuristic greedy coloring) can be implemented in a simple graph $G = (V, E)$ as follows:

1. Colourise vertices v_1 with color c_1.
2. For each vertices v_i in a row, selected the lowest numbered color that does not result in coloring is not valid (ie, such that no other vertices to v_i adjasen previously colored with the same color). Figure 4 illustrates the heuristic coloring on the same graph as in Figure 3.

Step 1. Select the maximum vertices degree, the vertices A given color 1 (red);

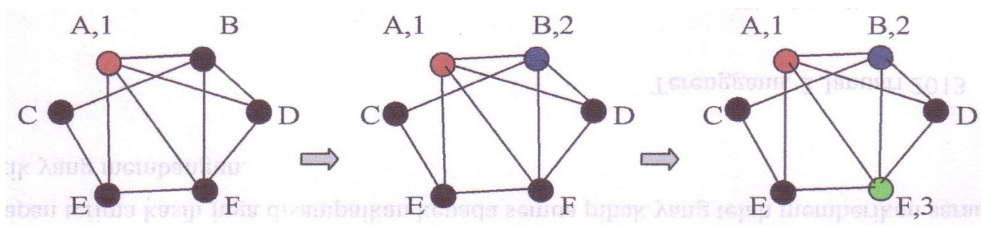
Step 2. Select the vertices that adjasen and degree less than or equal to A, that is vertices B are colored 2 (blue);

Step 3. Select the vertices that adjasen and degree less than or equal to A and B, the vertices F are color 3 (green);

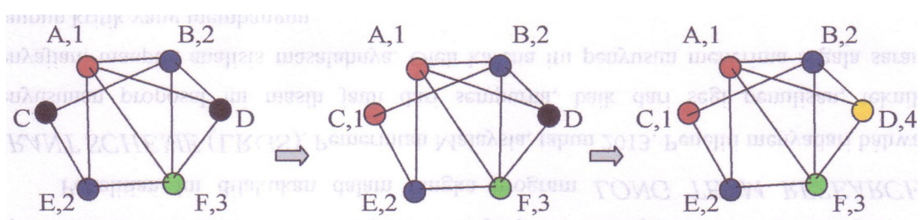
Step 4. Select the vertices that adjasen with F and A, namely vertices E are colored 2 (blue);

Step 5. Select the vertices that adjasen with E and B, the vertices C are color 1 (red);

Step 6. Last vertices *D* are colored 4 (yellow), because adjasen with color 1 vertices *A*, vertices *B* adjasen with color 2, and adjasen with vertices *F* color 3.



vertices A is colored 1 (red) vertices B is colored 2 (blue) vertices F is colored 3 (green)



vertices E is colored 2 (blue) vertices C is colored 1 (red) vertices D is colored 4 (yellow)

Figure 4. An illustration Coloring Heuristic Greedy, where $4 \leq \omega(G) \leq \chi(G) = 4$

The results of this heuristic is a valid coloring of *G*, which is the upper bound on $\omega(G)$, the size of the maximum clique in *G*.

Once the basics of graph theory has been reached, then to solve the problem on any graph of maximum clique appropriately measured using the branch and bound.

3.2 Branching Procedure

In this section of the procedure branching, using Figure 5, which describes the the branching following procedure:

1. Given graph $G = (V, E)$, selected branch vertices $v \in V$, where v is any vertices not connected to all other vertices in *G*. If there is no such v, then *G* is a clique. If not, then go to 2.

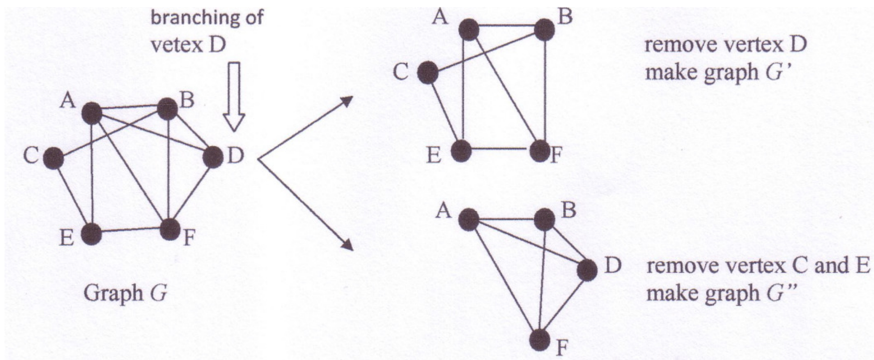


Figure 5. An Illustration of Branching Procedure

2. Create a graph of the G' and G'' of G as follows:
   - G' is the graph part of G induced by the vertices V-$\{v\}$, ie G' formed by deleting the vertices v (and adjacent edges) of G.
   - G'' is part of the graph G induced by vertices v in $N(v)$, where $N(v)$ indicates the neighbors of the vertices v, ie. G'' is formed by retaining only the vertices v and all vertices that adjasen to him in G.

Vertices maximum degree of a member or not a member of the maximum clique. Since removing G' only maximum vertices degree of a graph parent, and G'' maintain the maximum degree vertices and all its neighbors, the way in which the subgraph G' and G'' was made to ensure that a maximum clique in G will still be there in one or both of the subgraph. For more details, see [3, 4]. Thus, the maximum clique is not destroyed in this process, but the size of the graph at each the branching vertices decreases with each iteration.

4. Calculation Result of Branch and Bound

Now that the branching procedure and calculation boundary has been explained, the entire branch and bound algorithm can be further illustrated. Figure 6 shows the sequence of branching processes vertices, using upper and lower bounds to prune the branches, and the clique and coloring searched using heuristic described above. Clique is outlined in black, and colorings are shown generated using heuristic greedy clique. Clique alternatives can be sought when different options, to the maximum degree of a node is made (in the case of ties), and color alternatives may be sought if the order of vertices different from the one shown above.

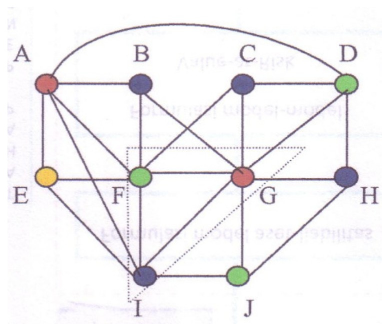


Figure 6. The sequence of branching processes vertices

Step 1. *Clique* = {F,G,I}; LB = 3 and Color = UB = 4

Branching on the vertices *G*

Step 2. Remove vertices G, obtained Clique = {A, E, F, I}, LB = 4. Color = UB = 4

Step 3. Remove all vertices that are not adjasen with G, obtained Clique = {F,G,I}; LB = 3. Color = UB = 3.

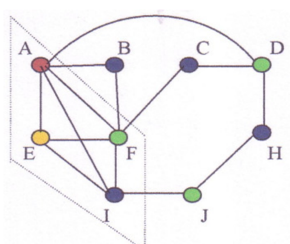


Figure 6. Step 2, Clique={A,E,F,I), LB=4. Color=UB=4

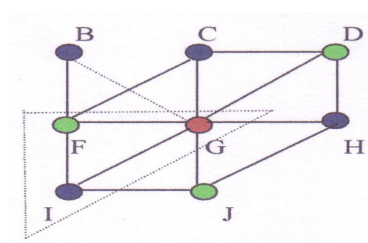


Figure 6. Step 3, Clique={F,G,I}, LB=3. Color=UB=4

Percabangan pada vertex F

Step 4. Remove vertices F in Step 2, obtained Clique = {A, E, I}; LB = 3. Color = UB = 4

Step 5. Remove all vertices that are not adjasen with vertices F in step 2, obtained Clique = {A, E, F, I}; LB = 4. Color = UB = 4.

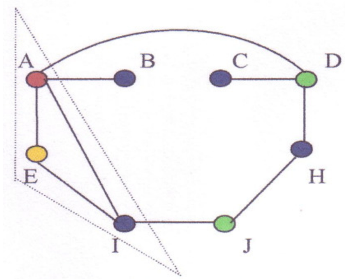


Figure 6. Step 4: Clique={A,E,I}, LB=3. Color=UB=3

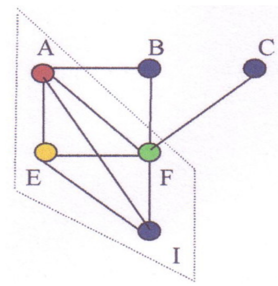


Figure 6. Step 5: Clique={A,C,D,G}, LB=4. Color=UB=4. Optimum

Figure 6, from step 1 to step 5, is an example of the whole procedure using branch and bound method in solving the maximum clique problem.

5. Conclusion

Branch and bound method has been used to the search the maximum clique problem. To determine the chromatic number $\chi(G)$of a graph G has been sought by the greedy heuristic, chromatic number $\chi(G)$is the upper bound (UB) on G and by branching procedure obtained lower bound (LB) is the maximum clique $\omega(G)$ on G. Results of the maximum clique the search also shows that the maximum clique $\omega(G)$ is less than or equal premises chromatic number $\chi(G)$. The use of branch and bound method is advantageous, because it provides an alternative perspective to solve the maximum clique problem, as well as the optimal method of solution requires no software optimization.

References

[1] Bomze, I. M., M. Budinich, P. M. Pardalos, M. Pelillo. 1999. The maximum clique problem. D.- Z. Du, P. M. Pardalos, eds. *Handbook of Combinatorial Optimization*, Vol. 4. Kluwer, Boston, 1–74.

[2] Ostergard, P.R.J. 2002. A Fast Algorithm for the Maximum Clique Problem. *Journal of Discrete Applied Mathematics,* 120 (2002) 197-207.

[3] Pardalos, P. M., J. Xue. 1994. The maximum clique problem. *J. Global Optim.* 4(3) 301–328.

[4] Strickland, D. M. 2002. Maximum cliques with application to protein structure alignment. Ph.D. thesis, School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta.

[5] Strickland, D. M. 2008. *Teaching Note—Using the Maximum Clique Problem to Motivate Branch-and-Bound.* INFORMS Transactions on Education 8(2), pp. 96–99.

[6] Stix, V. 2001. Finding all Maximal Clique in Dynamic Graphs. *Working Paper.* Vienna University of Economics, Department of Information Business, Austria.

[7] Tucker, A. 2002. *Applied Combinatorics*. Wiley, New York.

[8] Wilson, R. 1996. *Introduction to Graph Theory*. Pearson Education, Harlow, England.

[9] Winston, W. 2004. *Operations Research_ Applications and Algorithms*. Duxbury, Belmont, CA.

Received: October 21, 2013