Commun. Korean Math. Soc. 18 (2003), No. 4, pp. 745–754

A METHOD FOR COMPUTING UPPER BOUNDS ON THE SIZE OF A MAXIMUM CLIQUE

Koonchan Kim

Abstract. Maximum clique problem is to find a maximum clique (largest in size) in an undirected graph G. We present a method that computes either a maximum clique or an upper bound for the size of a maximum clique in G. We show that this method performs well on certain class of graphs and discuss the application of this method in a branch and bound algorithm for solving maximum clique problem, whose efficiency is depended on the computation of good upper bounds.

1. Introduction

We denote an undirected graph without loops or multiple edges by $G = (V, E)$, where $V = \{v_1, v_2, \ldots, v_n\}$ is the set of vertices and $E \subseteq V \times V$ is the set of edges. A *clique* of a graph G is a set of vertices, any two of which are adjacent, i.e., a complete subgraph of G. A *maximal* clique is a clique that is not a subset of any other clique. In *maximum clique* problem, one desires to obtain a maximum clique; a clique with the maximum cardinality, which is denoted by $\omega(G)$. The maximum clique problem is computationally equivalent to the *maximum independent (or stable) set* problem.

The maximum clique problem is difficult to solve and is known as NP-hard(see [4]), and hence no polynomial time algorithm is expected to be found. Most algorithms designed and developed for the maximum clique problem belong to one of the three categories: 1) enumerative algorithms that enumerate all of the cliques of a given graph, 2) implicit enumerative algorithms whose goal is to find one maximum clique, and 3) heuristics which compute an approximate solution to the maximum clique problem.

Received February 12, 2003.
2000 Mathematics Subject Classification: 05C69.
Key words and phrases: maximum clique problem, upper bound, branch and bound algorithm, clique, graph.

The most practical and commonly used algorithms for solving maximum clique problem are the *branch and bound* algorithms which belong to the categories of implicit enumerative algorithms. The key issues(see [9]) in a branch and bound algorithm are (1) how to find a good lower bound(a clique of large size), (2) how to find a good upper bound on the size of maximum clique for each subgraph generated, and (3) how to branch(selecting a vertex in a tree to form a new subgraph).

In the branch and bound algorithms of Garraghan and Pardalos [5] and Pardalos and Rodgers [8], the size of a given subgraph is used as an upper bound for the size of a maximum clique in order to prune the 'unprofitable' vertices. In recently proposed branch and bound algorithms of Balas and Xue [1], Östergard [7] and Wood [11], for example, the vertex coloring heuristic is used to obtain an upper bound for the size of a maximum clique for each subgraph generated. Here we state a vertex coloring heuristic described in Biggs [2] for later reference.

> COLOR: To determine a color class C_k, set $C_k = \emptyset$ and initialize S to be the set of uncolored vertices. While $S \neq \emptyset$, assign color k to a vertex $v \in S$ with maximum degree in G, and set $C_k := C_k \cup \{v\}$ and $S := (S \setminus \{v\}) \setminus N_G(v)$.

Note that if a graph G can be k-colored, i.e., if V is partitioned into k color classes $(C_1, C_2, \ldots, C_k)$, then k is an upper bound for $\omega(G)$, and consequently k becomes an upper bound for the size of maximum clique of G. Results on some theoretical upper bounds on the size of a maximum clique can be found in [3, 6].

The purpose of this paper is to introduce a method that computes either a maximum clique or an upper bound on the size of a maximum clique of a graph G and to study its characteristics. We discuss the application of this method in a branch and bound algorithm for solving maximum clique problem. Section 2 gives a condition for a number to be an upper bound, and Section 3 presents a criterion for the recognition of a maximum clique in a graph. Section 4 describes a method and illustrates the characteristics of this method on small representative examples. Section 5 concludes the paper.

2. An upper bound condition

Let $A(G) = (a_{ij})$ be the adjacency matrix for the graph $G = (V, E)$, where $a_{ij} = 1$ if $(v_i, v_j) \in E$ is an edge of G and $a_{ij} = 0$ if $(v_i, v_j) \notin E$ for $i, j = 1, 2, \ldots, n$. We assume that $E \neq \emptyset$; that is, G does not consist

K_m	tri(v)	deg(v)
K_1	0	0
K_2	0	1
K_3	1	2
K_4	3	3
K_5	6	4
K_6	10	5
$\vdots$	$\vdots$	$\vdots$
K_m	$\frac{(m-2)(m-1)}{2}$	$m-1$

Table 1. Number of triangles and degree formed by each vertex v in K_m

of only the isolated vertices, which implies that $\omega(G) \geq 2$. We denote $A^k(G)$ the kth power of $A(G)$ under matrix multiplication. Then the (i, j)-entry of $A^k(G)$ gives the number of walks of length k from vertex v_i to v_j in G. If we let t be the vector of the diagonal elements of $A^3(G)/2$, denoted by $t = diag(A^3(G)/2) = (t_1, t_2, \ldots, t_n)$, then the ith element t_i of t represents one half of the number of walks of length 3 from the vertex v_i to itself. In other words, t_i represents the number of triangles formed by the vertex v_i, without counting the same triangle twice. We denote this by $tri(v_i)$. We also denote the vector of the degrees of the vertices in G by $d = (d_1, d_2, \ldots, d_n)$, where $d_i = deg(v_i)$; the number of edges incident to v_i.

For example, let G be the $kite(K_4 \setminus \{e\})$, the complete graph of order 4 with an edge deleted. Then $d = (2\ 3\ 2\ 3)$ and $t = A^3(G)/2 = (1\ 2\ 1\ 2)$, in which v_1 and v_3 are the vertices with degree 2 and v_3 and v_4 are the vertices with degree 3.

In Table 1, the number of triangles and degree formed by each vertex v of the complete graph K_m are shown. Note that since any pair of vertices in K_m is adjacent, the number of triangles formed by a vertex v in K_m, without counting the same triangle twice, is the same as the number of ways of selecting two edges from the set of $m - 1$ edges that are incident to v, which is the combination $C(m - 1, 2)$ and is equal to $(m - 2)(m - 1)/2$.

From the Table 1, it can be easily seen that if a given graph G contains a clique of size m(i.e., a subgraph K_m), then there must exist at least m vertices in G such that the number of triangles formed by each of these vertices is at least $(m - 2)(m - 1)/2$.

An integer $m \geq 2$ is said to satisfy *upper bound condition* if it satisfies the following two conditions simultaneously:

1. G has a set of at least m vertices each of which forms at least $(m-2)(m-1)/2$ triangles
2. G does not have a set of $m+1$ vertices each of which forms greater than or equal to $(m-1)(m)/2$ triangles.

If there exists an integer m that satisfies the upper bound condition, then m becomes an *upper bound* for the size of a maximum clique of the graph G, which we denote by $\mu = m$. That is, $\omega(G) \leq \mu$.

For example, in the graph kite($G = K_4 \backslash \{e\}$), there are at least $m = 3$ vertices each of which forms at least $(m-2)(m-1)/2 = 1$ triangles(recall that $t = (1\ 2\ 1\ 2)$). However, there does not exist a set of $m+1 = 4$ vertices each of which has greater than or equal to $(m-1)(m)/2 = 3$ triangles. Thus, $m = 3$ satisfies the upper bound condition and hence $\mu = 3$ is an upper bound for the size of a maximum clique of the graph kite.

3. A criterion for the recognition of a maximum clique

Let $v \in V$ and $S \subseteq V$ for an undirected graph $G = (V, E)$. We denote $N_G(v)$ the set of vertices adjacent to v and $G(S)$ the subgraph of G induced by S. When there is no possibility of confusion, we will use the same symbol to denote both the vertex set of a clique and the clique itself.

Consider a complete graph K_m. Let v be a vertex of K_m. Then from Table 1, tri$(v) = (m-2)(m-1)/2$ and deg$(v) = m-1$. Conversely, we have the following result:

LEMMA 1. *Let $G = (V, E)$ be an undirected graph with $E \neq \emptyset$. If there exists a vertex v of G and an integer $m \geq 2$ such that tri$(v) = (m-2)(m-1)/2$ and deg$(v) = m-1$, then $C = N_G(v) \cup \{v\}$ forms a maximal clique of G.*

PROOF. Suppose there exists a vertex v of G and an $m \geq 2$ such that tri$(v) = (m-2)(m-1)/2$ and deg$(v) = m-1$. Since deg$(v) = m-1$, there are $m-1$ vertices adjacent to v. Let $C = N_G(v) \cup \{v\}$. We show that C is a clique, a complete subgraph K_m of G. Let u be an arbitrary vertex in $N_G(v)$. To show C is a clique, we must show that u is adjacent to all of the other vertices in $N_G(v)$. But this result follows immediately since the cardinality of $N_G(v)$ is $m-1$ and the maximum number of triangles that the vertex v can form using the vertex u(i.e., edge (v, u))

and one of the vertices in $N_G(v)$ is $m-2$; if there exists a vertex w in $N_G(v)$ such that u and w are not adjacent, then tri(v) would not equal to $(m-2)(m-1)/2$, which is a contradiction. Also, it can be easily seen that no other clique could contain C as a proper subset(as a proper subgraph), so C is a maximal clique of G. □

Next we present a *criterion* for the recognition of a maximum clique in a graph G.

PROPOSITION 2. *Let $G = (V, E)$ be an undirected graph with $E \neq \emptyset$. Suppose that there exists a vertex v in G and an integer $m \geq 2$ such that tri$(v) = (m-2)(m-1)/2$, deg$(v) = m-1$, and m satisfies the upper bound condition. Then the set $C = N_G(v) \cup \{v\}$ forms a maximum clique of G and $\omega(G) = m$.*

PROOF. By Lemma 1, the first two hypothesis implies that C is a maximal clique with size m. Since m satisfies the upper bound condition, m is an upper bound for the maximum clique of G. Consequently, C must be a maximum clique of G and $\omega(G) = m$. □

REMARK. Proposition 2 implies that if there exists a vertex v such that deg$(v) > m-1$ but tri$(v) = (m-2)(m-1)/2$, then v can not belong to a clique of size m.

In the rest of this paper when we say 'the criterion' we mean the criterion given in Proposition 2.

EXAMPLE 1. Consider the graph G given in Figure 1.

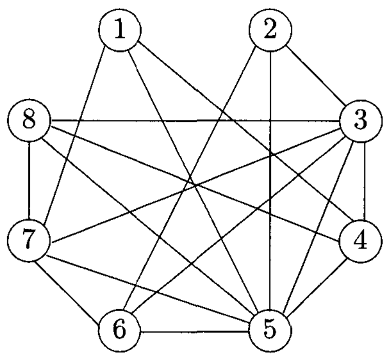

Figure 1. G

Computing the number of triangles and degree formed by each vertex in G, we obtain, respectively,

$$\begin{aligned} t &= (2\ 3\ 9\ 4\ 11\ 5\ 6\ 5) \\ d &= (3\ 3\ 6\ 4\ 7\ 4\ 5\ 4). \end{aligned}$$

Since there exists at least four vertices(all except one vertex) each of which has at least three triangles and no set of five vertices exists with each vertex forming greater than or equal to six triangles (only three vertices have greater than or equal to six triangles), it follows that $\mu = 4$ is an upper bound for the size of a maximum clique in G. Furthermore, the criterion is satisfied at $v_2(\text{tri}(v_2) = 3$ and $\deg(v_2) = 3)$. Thus, $\omega(G) = \mu = 4$ and $C = \{v_2, v_3, v_5, v_6\}$, which is equal to $N_G(v_2) \cup \{v_2\}$, forms a maximum clique of G.

4. A method

Before we present a method, we describe how a tighter(better) upper bound can be obtained. A tighter upper bound can be computed by removing the so-called 'unprofitable' vertices; vertices that do not have enough triangles.

Let $V = \{v_1, v_2, \ldots, v_n\}$ be the set of vertices of G. Compute $t = \text{diag}(A^3(G)/2)$ and obtain an upper bound $\mu = m$. If the criterion is satisfied at a vertex, then extract a maximum clique and stop. If not, remove all the vertices with triangles fewer than $(m-2)(m-1)/2$ from V and compute t corresponding to the new set of vertices. Repeat this removal process until one of the following two cases occurs:

1. no further removal of vertices is possible; in this case, the number of vertices(vertices with triangles greater than or equal to $(m-2)(m-1)/2)$ is greater than or equal to m.
2. the number of vertices(vertices with triangles greater than or equal to $(m-2)(m-1)/2)$ becomes less than m.

If the first case occurs, then m is an upper bound for the size of a maximum clique of G and we are done. If the second case occurs, then m can not be the size of a maximum clique of G. Obviously then $\mu = m-1$ is a better upper bound. With $m-1$ as a new tighter upper bound, one can then repeat the above removal process on the original set V to see if even a tighter upper bound can be obtained; i.e., remove all the vertices from V with triangles fewer than $(m-3)(m-2)/2$, etc.

When computing a maximum clique or an upper bound, note that any vertex v which is adjacent to all of the other vertices in V can be eliminated since $\omega(G) = \omega(G \setminus \{v\}) + 1$.

We now formulate a method which we call UBT(upper bound based on triangles).

A Method(UBT):

1. Given G, compute $t = \mathrm{diag}(A^3(G)/2)$, d, and an upper bound m.

2. Find the most tight upper bound μ using the procedure described above. Each time new t is computed, check if the criterion is satisfied, and if a vertex satisfies the criterion, then extract a maximum clique and stop.

3. Accept μ as an upper bound for the size of a maximum clique of G.

EXAMPLE 2. Consider the graph G' given in Figure 2, which is obtained by elimininating two edges (v_3, v_5) and (v_3, v_7) from G in Figure 1.

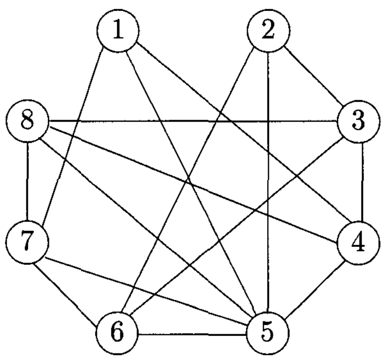

Figure 2. $G' = G \setminus \{(v_3, v_5) \cup (v_3, v_7)\}$

Computing the number of triangles and degrees, we get

$$\begin{aligned} t &= (2\ 2\ 2\ 3\ 6\ 3\ 3\ 3) \\ d &= (3\ 3\ 4\ 4\ 6\ 4\ 4\ 4). \end{aligned}$$

As can be seen $m = 4$ becomes an upper bound, but no vertex satisfies the criterion. Remove the vertices v_1, v_2, and v_3 from V and let $V_1 = \{v_4, v_5, v_6, v_7, v_8\}$. Let H be the subgraph induced by V_1 and $A(H)$ be the adjacency matrix. Computing the number of triangles corresponding to the vertices in V_1 gives

$$t_1 \;=\; \mathrm{diag}(A^3(H)/2) = (1\ 3\ 1\ 2\ 2).$$

Since the second case has been occurred(only one vertex has triangles greater than or equal to 3), it follows that $m = 4$ can not be the size of a maximum clique in G'. We let $m = 3$ be the new tighter upper bound and we repeat the above process on the original set V. However, as can be observed no further removal of vertices is possible. Also, no vertex in V satisfies the criterion. Thus, UBT yields $\mu = 3$ as the most tight(best) upper bound for the size of a maximum clique of G'. Note that $\omega(G') = 3$.

The next example considers a class of graphs, so-called triangle-free graphs, on which UBT performs well.

EXAMPLE 3. Let $G = (V, E)$ be an undirected graph with $E \neq \emptyset$. Suppose it turns out that $t = \mathrm{diag}(A^3(G)/2) = 0$. Then from Table 1, an upper bound is either 0 or 1. Since $E \neq \emptyset$, one can then immediately conclude that $\omega(G) = 2$ and any edge can be a maximum clique of G. Examples of such graphs include the Peterson graph, n-cycle graphs $C_n (n \geq 5)$, bipartite graphs, graphs generated by Mycielski's construction(these graphs have large chromatic numbers), etc.

5. Discussions and remarks

Most of the upper bound techniques employed in the branch and bound algorithms are based on the vertex coloring heuristics. Vertex coloring heuristics works well on many instances, but there exist graphs with no triangle but with large chromatic numbers. On such class of graphs, vertex coloring heuristics usually yields 'loose' upper bounds. For example, Grötzsch graph(constructed by the Myciekski's method(see [10], p.205)) is triangle-free and is of order 11. When applying the vertex coloring heuristic(COLOR) stated in Section 1, we obtain $\mu = 4$ as an upper bound(its chromatic number is also 4).

For UBT, each computation of $A^3(G)/2$ requires $O(n^3)$ arithmetic operations, which is rather expensive, and for dense graphs it often yields loose upper bounds. However, on many instances such as graphs with triangle-free(as demonstrated in the previous section), sparse graphs, graphs with not much triangles, and so on, UBT performs well. Hence, both techniques UBT and vertex coloring heuristics exhibit merits and demerits. It would be advantageous then to combine them together and use it as an upper bound technique in a branch and bound algorithm. This approach and the development of UBT into a practical method for solving maximum clique problem are the subject of future study.

Finally, we note that a much better upper bound can also be computed by considering each subset $S_i = N_G(v_i) \cup \{v_i\}$ and finding a maximum clique or an upper bound on each S_i for $i = 1, 2, \dots, n$. In this case, the largest maximum clique among all the maximum cliques computed from the subproblems becomes a lower bound and the largest upper bound among all the upper bounds computed from the subproblems becomes an upper bound for the size of the maximum clique of G. Then, vertices corresponding to the upper bounds and the size of maximum cliques not larger than the size of the largest maximum clique (the lower bound) can be removed, and the removal process can be continued to obtain a better lower and upper bounds.

References

[1] E. Balas and J. Xue, *Weighted and unweighted maximum clique algorithms with upper bounds from fractional coloring*, Algorithmica **15** (1996), 397–412.

[2] N. Biggs, *Some Heuristics for Graph Coloring*, in: R. Nelson, R.J. Wilson, (Eds.), Graph Colourings, Longman, New York, (1980), pp. 87-96.

[3] A. Billionnet, *An upper bound on the size of the largest cliques in a graph*, Journal of Graph Theory **5** (1981), 165–169.

[4] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, San Francisco, 1979.

[5] R. Garraghan and P. M. Pardalos, *An exact algorithm for the maximum clique problem*, Operations Research Letters **9** (1990), 375–382.

[6] D. P. Geoffroy and D. P. Sumner, *An upper bound on the size of a largest clique in a graph*, Journal of Graph Theory, **2** (1978), 223–230.

[7] P. R. J. Östergard, *A fast algorithm for the maximum clique problem*, Discrete Applied Mathematics **120** (2002), 197–207.

[8] P. M. Pardalos and G. P. Rodgers, *A branch and bound algorithm for the maximum clique problem*, Comput. Oper. Res. **19** (1992), no. 5, 363–375.

[9] P. M. Pardalos and J. Xue, *The maximum clique problem*, Journal of Global Optimization **4** (1994), 301–328.

[10] D. B. West, *Introduction to Graph Theory*, Prentice Hall, 2001.

[11] D. R. Wood, *An algorithm for finding a maximum clique in a graph*, Operations Research Letters **21** (1997), 211–217.

Department of Mathematics
Keimyung University
Daegu 704-701, Korea
E-mail: kmkim@kmu.ac.kr