

Exact Polynomial-time Algorithm for the Clique Problem and $P = NP$ for Clique Problem

Kanak Chandra Bora

Department of Computer Science & Engineering
Royal School of Engineering & Technology,
Betkuchi, Guwahati-781035, Assam, India.

Bichitra Kalita

Department of Computer Application (M.C.A.),
Assam Engineering College,
Jalukbari, Guwahati-781013, Assam, India.

ABSTRACT

In this paper, the Minimum Nil Sweeper Algorithm, applicable to Clique problem has been considered. It has been found that the Minimum Nil Sweeper Algorithm is not applicable to Clique problem for all undirected graphs which was previously claimed. A new algorithm has been developed to study the all clique problems for arbitrary undirected graph and its complexity is analysed. An experimental result is cited. Finally, the $P = NP$ has been proved for Clique problem. A theorem related to intersection graph is developed.

General Terms

$P=NP$ for Clique

Keywords

Exact Polynomial-time Algorithm, Clique, Euler-diagram, Complexity. $P=NP$

1. INTRODUCTION

Based on solving time, problems are divided into three categories [1]. The class P consists of those problems that are solvable in polynomial time. Intractable problems are those which are unsolvable by polynomial-time algorithms. NP problems are those problems that have no polynomial-time algorithm. It has not yet been found [2], even though many attempts have made.

The concept of NP -complete problem had been introduced by S. A. Cook in 1971 [3]. A problem C is NP -complete [4] if it satisfies the following conditions

- (a). $C \in NP$, and
- (b). $A \leq_p C$ for any problem $A \in NP$.

The most important open problem in theoretical computer science is whether $P \neq NP$ or $P=NP$? This question has been one of the deepest, most perplexing open research problems in theoretical computer science since it was first posed in 1971 [3]. It is sufficient to present a polynomial time algorithm for any NP -complete problems [4][5]. Here Clique problem is considered and this is one of Richard Karp's 21 problems [6].

A clique in an undirected graph $G = (V, E)$ is a complete sub graph of the graph G . The size of a clique is the number of vertices it contains. The clique problem is one of the optimization problems of finding a clique of maximum size in a graph.

The clique problem is NP -complete since we can check in polynomial time whether some vertices of a graph form a clique and $3\text{-CNF-SAT} \leq_p \text{CLIQUE}$. Clique problem has many applications like in social network, bioinformatics and in computational chemistry [7].

K. Makino and T. Uno discussed the enumeration of maximal bipartite cliques in a bipartite graph [8]. E. Tomita et al discussed the worst-case time complexity for generating maximal cliques of an undirected graph [9]. Takeaki UNO explained techniques for obtaining efficient clique enumeration implementations [10].

Zohreh O. Akbari studied the clique problem and presented "The Minimum Nil Sweeper Algorithm", a deterministic polynomial time algorithm for the problem of finding the maximum clique in an arbitrary undirected graph [11]. The Minimum Nil sweeper algorithm was made considering all zeroes on the inaccessibility matrix. The basic idea behind the algorithm was that the problem of omitting the minimum number of vertices from the graph so that no zero would remain except the main diagonal in the adjacency matrix resulting sub graph.

Further, it has been proposed that, considering an arbitrary undirected graph, maximum clique size of the graph can be found using the "The Minimum Nil Sweeper Algorithm".[11]

It is known that heuristic algorithms can determine a problem quickly but they are not guaranteed to give a definite solution to all problems. Exact algorithms can determine a problem and they are guaranteed to give a definite solution to all problems.

The paper is organized in three sections. Section 1 presents introduction part containing previous works of other researchers including the Minimum Nil Sweeper Algorithm. Section 2 presents a new general algorithm to solve the Clique problem for an arbitrary undirected graph. Complexity of the general algorithm has been analysed and finally $P=NP$ for Clique problem is proved. In section 3, some examples graphs cited and are tested by both the algorithms. And finally a theorem related to the intersection graph is stated.

2. GENERAL ALGORITHM & IT'S COMPLEXITY

2.1 General Algorithm: One general algorithm is presented here for finding the maximum clique size in an arbitrary undirected graph. In the adjacency matrix of an undirected graph, the elements which are 1, are only considered. In addition to this, row wise clique size is considered and finally maximum clique size is found. If the given graph is a complete graph with n vertices then the maximum clique size of the graph is $n-1$. If there is only one vertex in the graph then clique size is 1 and if there are two vertices in the graph

which are connected then the clique size is 2 and for these two cases our algorithm is not required to apply.

Algorithm: we construct an adjacency matrix A_{ij} for the given graph whose total number of vertices are n . $T, i, j, p, w, x, y, K, R, \text{Big}$ are integer variables. $B[], \text{Clcount}[], \text{CL}[], \text{BClique}[], B_{ij}, \text{Clique}[]$ are array of type integer.

// make the adjacency matrix

$A_{ij} = 1$ if there is an edge from V_i to V_j

For ($i = 1; i \leq n; i++$)

{
For($j = 1; j \leq n; j++$)

{

 $A_{ij} = 0$ or $1;$

}

}

For ($i = 1; i \leq n; i++$)

{
for ($j = 1; j \leq n; j++$)
{
While ($i \neq j$)
{
If ($A_{ij} \neq 1$)
Goto Levelst;
}
}
}

}
Printf(" Maximum clique size is = $n-1$ ");

Goto Last;

Levelst: For ($i = 1; i \leq n; i++$)

{
 $T = 1;$
For ($j = 1; j \leq n; j++$)
{
If ($A_{ij} == 0$)
Continue;
 $B[T] = j;$
 $T = T + 1;$
}
 $K = 0;$
LevelB: For ($w = 1; w \leq T - 2; w++$)
{
 $\text{Clcount}[w] = 0;$
 $R = 1;$
 $\text{CL}[R] = B[w];$
For ($p = w; p \leq T-2; p++$)
{
If ($A_{B[w]B[p+1]} == 1$)
{

$\text{Clcount}[w]++;$
 $\text{CL}[++R] = B[p + 1];$

}

}

IF($\text{Clcount}[w] == 0$)

{

$\text{BClique}[w] = 0;$

If($w == T-2$)

{

Goto LevelBig;

}

Goto LevelB;

}

IF ($\text{Clcount}[w] > 0$)

{

If ($\text{Clcount}[w] == 1$)

{

$\text{BClique}[w] = 3;$

If($w == T-2$)

{

Goto LevelBig;

}

Else

Goto LevelB;

}

Else

{

LevelA: For ($x = 1; x \leq \text{Clcount}[w] + 1; x++$)

{

$R = x;$

$\text{CL}[0] = 0;$

$Z = \text{CL}[x-1];$

For ($y = x + 1; y \leq \text{Clcount}[w] + 2; y++$)

{

If ($x = 1$ and $y \neq 1$)

{

$K = \text{CL}[R];$

$B_{xy} = A_{iK};$

$R++;$

If($B_{xy} == 0$)

{

$\text{BClique}[w] = 0;$

If($w == T-2$)

{

Goto LevelBig;

}

Else

{

Goto LevelB;

}

}

}

If ($x \neq 1$ and $y \neq 1$)

{

$K = \text{CL}[R];$

$B_{xy} = A_{ZK};$

$R++;$

If($B_{xy} == 0$)

```

{
  BClique[w] = 0;
  If( w == T-2)
  {
    Goto LevelBig;
  }
  Else
  {
    Goto LevelB;
  }
}
}
}
}
BClique[w] = Clcount [w] + 2 ;
}

LevelBig:Big = BClique[1];
W = T-2
For (u = 1; u <= W; u++)
{
  If (Big <= BClique[u])
    Big = BClique[u];
}
Clique[i] = Big ;
}
Big = Clique[1];
For (i = 1; i < n; i++)
{
  If(Big < Clique[i])
    Big = Clique[i];
}
If (Big==0)
Big=2;
Printf("Maximum clique size =
Big ");
Last:      STOP END.

```

2.2 Complexity Analysis of the Algorithm:

Suppose number of vertices in the undirected graph is 'n', then complexity of the algorithm will be as follows
 $= n^2 + n^2 + 1 + n[n + (T-2)\{(T-2) + (\text{count} + 1)^2\} + n] + n$
 $= 2n^2 + 1 + n^2 + n(T-2)^2 + n(T-2)^2 + n(T-2)(\text{count} + 1)^2 + n^2 + n$
 Here "T" and "count" can be considered as (n- something) , something < n, and by which complexity of the algorithm will be $= O(n^4)$

Since Clique problem is an NP-complete problem and from the above General Algorithm[2.1], it is proved that Clique problem can be solved in polynomial time, hence $P = NP$ for Clique problem.

3. TESTING OF ALGORITHM FOR GRAPH

3.1 Case-1:

Let us verify the algorithms for the graph of figure-1

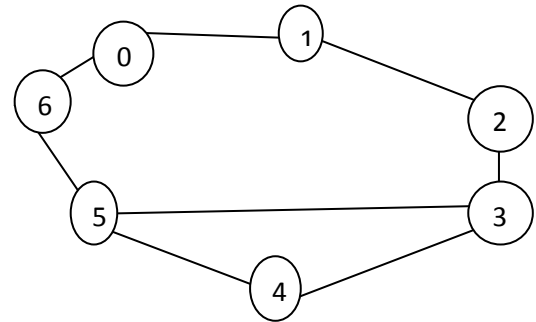


Figure 1

Minimum Nil Sweeper Algorithm gives the out put for the graph figure 1, which shows that the vertices 5, 6, forms a clique, which is not a maximum clique. After application of the General Algorithm [2.1] , the maximum clique size = 3, is found for the graph of figure 1, which is the correct result.

3.2 Case-2:

Let us consider the graph of figure-2.

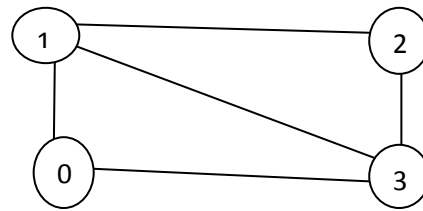


Figure 2

Again when the Minimum Nil Sweeper Algorithm is used, then the vertices 1, 2, 3 forms a maximum clique of size 3. The same result is obtained after application of the General Algorithm[2.1] .

3.3 Case-3:

Let us consider the graph of figure-3.

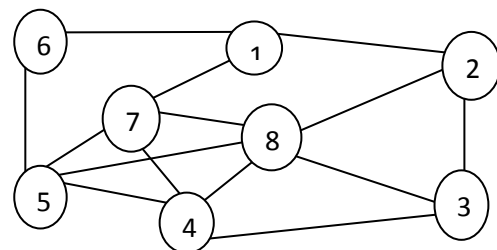


Figure 3

For the graph of figure 3, the maximum clique of size 4 is obtained after application of the Ggeneral Algorithm [2.1] .

3.4 Case-4:

Let us consider the graph of figure-4. This graph is considered from a paper of reference number [12], which is known as intersection graph.

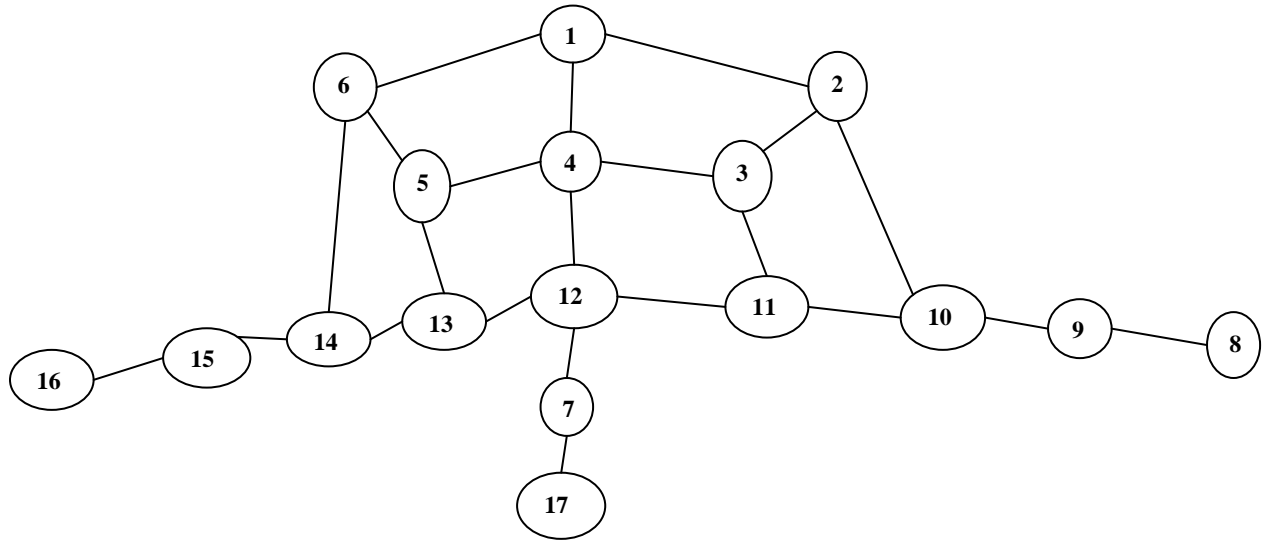


Figure 4

Now applying the General Algorithm, it is found that maximum clique size = 2. Hence the following theoretical result has been established.

Theorem: There does not exist clique size greater than 2 for any intersection graph obtained from a class of Euler diagram [12]

Proof: It is found that the intersection graph does not contain any closed circuit in which no pairwise adjacent vertices exists. On the other hand, it has been established [9] that if one go on adding number of curves horizontally as shown in figure-5, which was discussed in [12].

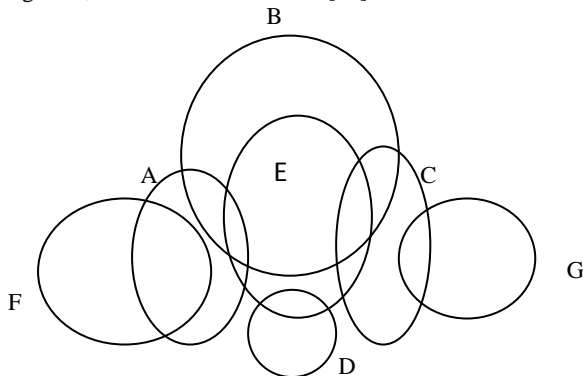


Figure 5

This process definitely increases the number of curves horizontally and the intersection graph is obtained as discussed in [12]. Hence no closed circuit exist here as stated above and this completes the proof of the theorem.

4. COMPARATIVE ANALYSIS OF EXISTING TECHNIQUES

Bron et al [13] implemented a depth-first search algorithm for generating all maximal cliques of a graph and the computing time of the algorithm was proportional to $(3.14)^{n/3}$ for Moon-Moser graph [14] of n vertices. Later on E. Tomita et al proposed a depth-first search algorithm for generating all maximal cliques of an undirected graph as in Bron-Kerbosch algorithm. The worst-case time complexity was $O(3^{n/3})$, where n is the number of vertices in a graph. Tsukiyama et al proposed another algorithm for enumerating a maximal independent sets of a graph in $O(nm\mu)$ - time, where n, m , and μ are the number of vertices, edges, and maximal independent sets of the graph respectively [15]. The algorithm of Tsukiyama et al was improved by Chiba and Nishizeki and the complexity of the algorithm became $O(a(G)m\mu)$, where $a(G)$ is the arboricity of G with $a(G) \leq O(m^{1/2})$ for a connected graph G and μ is the number of maximal cliques in graph G [16]. K. Makino et al discussed two algorithms for finding maximal bipartite cliques of a bipartite graph. One runs with $O(M(n))$ time delay and the other runs with $O(\Delta^4)$ time delay taking preprocessing time as $O(nm)$, where Δ is the maximum degree of the graph, and $M(n)$ is the time needed to multiply two $n \times n$ matrices, n is the number of vertices and m is the number of edges of the graph G . Takeaki UNO

proposed algorithm which takes polynomial delay at most $O(|V|)$ time for each clique and $O(|V||E|)$ time for each maximal clique.

From the above existing algorithms it is observed that some algorithms complexity depends on the maximum degree of the graph, some algorithms complexity depends on the number of edges and vertices of the graph, some algorithms complexity depends on arboricity of G and maximal cliques in G and some algorithms can be applied only for bipartite graph. Therefore a theoretical comparison of these algorithms is found to be difficult with the General Algorithm 2.1, discussed here, as this algorithm is applicable for any pattern of undirected graph in polynomial time.

5. CONCLUSION

From the above examples [3.1 case-1 & 3.2 case 2] it is found that the Minimum Nil Sweeper Algorithm cannot be applied to find the maximum clique size in an arbitrary undirected graph. On the other hand the General Algorithm established here can be applied to find the maximum clique size in an undirected arbitrary graph.

6. ACKNOWLEDGMENTS

We gratefully thank to the referee for giving an advice to improve the paper in present form

7. REFERENCES

- [1] Richard E. Neapolitan and Kumarss Naimipour, "Foundations of Algorithms using C++ Pseudocode", 3rd ed, Jones and Bartlett Publishers, 2003, ch. 9.
- [2] Michael Sipser, "Introduction to the Theory of Computation", 2nd ed., International Edition, Thomson Course Technology, p 270, definition 7.19 and theorem 7.20, 2006.
- [3] Stephen A. Cook, "The complexity of theorem-proving procedures", Proceedings of Third Annual ACM Symposium on Theory of Computing, pages 151-158, 1971.
- [4] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein, "Introduction to Algorithms", 2nd ed., MIT Press and McGraw-Hill, 2001, ch. 22 and ch. 34.
- [5] Stephen A. Cook, "The P versus NP problem", Manuscript prepared for the Clay Mathematics Institute for the Millennium Prize Problems, 2000.
- [6] Richard M. Karp, "Reducibility among combinational problems", In R. E. Miller and J. W. Thatcher (editors): Complexity of Computer Computations, pages 85-103. New York: Plenum Press, 1972.
- [7] Alon, N.; Boppana, R. (1987), "The monotone circuit complexity of Boolean functions", *Combinatorica* 7(1): 1-22, doi:10.1007/BF02579196.
- [8] K. Makino and T. Uno, "New algorithms for enumerating all maximal cliques," In Proc. of the 9th Scandinavian Workshop on Algorithm Theory (SWAT 2004), pp.260-272. Springer-Verlag, 2004
- [9] E. Tomita, A. Tanaka, and H. Takahashi, "The worst-case time complexity for generating all maximal cliques and computational experiments," *Theoretical Computer Science*, vol.363, pp.28-42, 2006.
- [10] Takeaki UNO, "Implementation issues of clique enumeration algorithm", Special issue: Theoretical computer science and discrete mathematics, *Progress in Informatics*, No.9, pp.25-30, (2012).
- [11] Zohreh O. Akbari, "A Deterministic Polynomial-time Algorithm for the Clique Problem and the Equality of P and NP Complexity Classes", *World Academy of Science, Engineering and Technology* 45 2008.
- [12] Rongdeep Pathok, Bichitra Kalita, "Properties of Some Euler Graphs Constructed from Euler Diagram", *Int. Journal of Applied Sciences and Engineering Research*, Vol. I, No. 2, 2012, pp.232-237.
- [13] C. Bron, J. Kerbosch, Algorithm 457, finding all cliques of an undirected graph, *Comm. ACM* 16 (1973) 575-577.
- [14] J. W. Moon, L. Moser, On cliques in graphs, *Israel J. Math.* 3 (1965) 23-28.
- [15] S. Tsukiyama, M. Ide, H. Ariyoshi, I. Shirakawa, A new algorithm for generating all the maximal independent sets, *SIAM J. Comput.* 6 (1977) 505-517.
- [16] N. Chiba, T. Nishizeki, Arboricity and subgraph listing algorithms, *SIAJ J. Comput.* 14 (1985) 210 – 223.