Proceedings of the International Conference on Industrial Engineering and Operations Management
Riyadh, Saudi Arabia, November 26-28, 2019

An Enumerative Algorithm for the Maximum Clique Problem

Mochamad Suyudi, Sukono
Department of Mathematics, Faculty of Mathematics and Natural Sciences
Universitas Padjadjaran, Indonesia
moch.suyudi@gmail.com, sukono@unpad.ac.id

Mustafa Mamat
Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin
Tembila Campus, 2200 Besut, Terengganu, Malaysia
musmat567@gmail.com

Abdul Talib Bon
Department of Production and Operations,
University Tun Hussein Onn Malaysia, Johor, Malaysia
talibon@gmail.com

Abstract

Partial enumerative algorithms are presented for maximum clique problems which are very easy to implement. Search results for efficient implementation of any graph for randomly generated graphs of up to 10 vertices and more than ten sides. Appropriate specifications are also provided for testing problems to facilitate future comparisons. In addition, to understand the maximum clique search without using any software. Search results are presented in the form of examples.

Keyword: maximum clique, enumerative algorithm, current best clique.

1. Introduction

Given a graph $G(V, E)$, where V is the set of vertices and E is the set of edges, a complete subgraph of G is one whose vertices are pairwise adjacent. The *maximum clique problem* is to find the maximum complete subgraph of G. An independent set is a set of vertices whose elements are pairwise nonadjacent. The *maximum independent set problem* is to find the largest such indepent set. A vertex cover is a set of vertices that covers all the edges of G. The *minimum vertex cover problem* is to find the minimum of such set. These three problems are computationally equivalent. They are also known to be NP-complete (P. Berman and G. Schnitger, 1989)2]).

The maximum clique problem has numerous applications in science and engineering(E. Balas and C.S. Yu, 1986) [1]. One way to cope with an NP-complete problem is to design an algorithm that performs fast on the average with respect to a natural probability distribution of inputs. Most of the algorithms for the maximum clique problem degenerate to an exhaustive search. An obvious enumerative algorithm is to consider all subsets of V in decreasing order of cardinality and check if each one forms a clique. Stop the checking when such a set has been found. Algorithms for the maximum clique problem have been studied by many authors (E. Balas and C.S. Yu, 1986), (C. Bron and J. Kerbosch, 1973) and (R.E. Tarjan and A.E. Trojanowski, 1977). Most of the proposed algorithms have been implemented and tested on medium size problems using random graphs. Recently, using a

branch and bound method based on quadratic 0-1 programming (P.M. Pardaios and G.P. Rodgers, 1990), problems with 1000 vertices and as many as 150000 edges were solved.

In this paper, we propose a simple and very efficient algorithm for finding the maximum clique in a graph *G*. The algorithm uses a partial enumeration and is very simple to implement. Computational experience with randomly generated graphs shows that the algorithm is faster than any previous known method.

2. Algorithm description

Initially, the algorithm considers an ordering of the vertices of *G*, say $v_1, v_2, \ldots, v_n$, where v_1 is a vertex of smallest degree in *G*, v_2 is a vertex of smallest degree in $G-(v_1)$, and generally v_k is a vertex of smallest degree in $G-\{v_1, v_2, \ldots, v_{k-1}\}$, $k \leq n-2$.

It has been observed that this vertex ordering reduces the computational time in problems withdense graphs. In addition, this ordering can be reapplied at depths > 1. For lower density problems the algorithm is faster when no ordering of the vertices is performed.

Without any pruning, the algorithm produces a most *n* cliques of increasing size. Initially, it finds the largest clique C_1 that contains the vertex v_r. Then it finds C_2, the largest clique in $G-(v_1)$, that contains v_2 and so on. Applying heuristics and pruning techniques, we can reduce the search space dramatically.

Crucial to the understanding of the algorithm is the notion of the *depth.* Suppose we consider vertex v_1. At depth 2, we consider all vertices adjacent to v_1 . At depth 3, we consider all vertices (that are in depth 2) adjacent to the first vertex in depth 2 (the example below makes this dear). Let v_{di} be the vertex we are currently expanding at depth *d* and step *i*. That is,

$$\text{Depth } d\text{: } v_{d1} \ldots . \, v_{di} \ldots . \, , \, v_{dm}.$$

If $d + (m - i) \leq$ size of current best clique (CBC), then we prune, since the size of the largest possible clique (formed by expanding v_{di}) would be less or equal to the size of CBC. If we are at depth 1 and this inequality holds then we stop; we have found the maximum clique (R. Carraghan and P. M. Pardalos, 1990).

If it is known that the maximum clique has size $\geq \alpha$, then we may use as pruning (stopping) criterion the following:

$$d + (m - i) \leq \alpha.$$

If α is close to the size of the actual maximum clique, the computational time can be reduced

dramatically for graphs with high density. In (P. Turan, 1954) proved that every graph with *n* vertices and *m* edges contains a clique of size $\geq n^2/(n^2 - 2m)$.

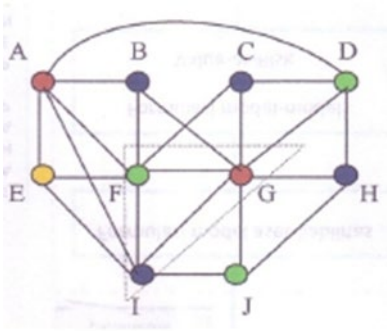
Hence, if the graph has density *D* then we may take

$$\alpha = \frac{n}{(1-D)n + D}.$$

Unfortunately, the value of a is helpful only if it is very close to the actual maximum clique size. However, polynomial time algorithms can be used to find very good values for α (E. Balas and C.S. Yu, 1986).

3. Result

In this section we present results to show the efficiency of our algorithm. Consider the graph in Figure 1, with the steps of the algorithm shown in Table 1. We have found the maximum clique for this graph {A, E, F, I}, with size 4.



Figur 1. Vertex Coloring Graph.

Table 1. Vertex Coloring

1	Initialization orders the vertices	H J B C E D A I F G	
2	Depth l:	**H** J B C E D A I F G	
	Depth 2:	**J** D G	
	Depth 3:	**G**	Cannot expand, so CBC is {H, J, G}, size 3
	Depth 2:	**J** D G	Since $d + (m - i) = 2 + (3 - 2) = 3 \leq 3,$ we prune.
	We are now finished with vertex H		The largest clique containing this node is {H, J, G}
3	Depth l:	H **J** B C E D A I F G	
	Depth 2:	**I** G	2+(2-1) ≤CBC, so prune
4	Depth l:	H J **B** C E D A I F G	
	Depth 2:	**A** F G	
	Depth 3:	**F**	Cannot expand, so CBC is {B, A, F}, size 3
	Depth 2:	**A** F G	2+(3-2) ≤CBC, so prune
5	Depth l:	H J B **C** E D A I F G	
	Depth 2:	**D** F G	
	Depth 3:	**G**	Cannot expand, so CBC is {C, D, G}, size 3
	Depth 2:	**D** F G	2+(3-2) ≤CBC, so prune
6	Depth l:	H J B C **E** D A I F G	
	Depth 2:	**A** F I	
	Depth 3:	**F** I	
	Depth 4:	**I**	{E, A, F, I} becomes our new CBC of size 4
	Depth 3:	**F** I	3+(2-1) ≤ 4, so prune
	Depth 2:	**A** F I	2+(3-2) ≤ CBC, so prune
7	Depth l:	H J B C E **D** A I F G	1+(10-4) ≥ CBC and depth 1, so stop

We have the maximum clique for this graph {A, E, F, I}, with size 4 of step 6.

4. Concluding remarks

We have shown that a simple algorithm implemented very efficiently can solve very large maximum clique problems. Since this algorithm is very simple to implement, it can be used as a basis for computational comparison when different algorithms are used. The proposed algorithm can be easily parallellized, since vertex v_{1k} (depth 1) can be processed independently of vertices $v_{11}, \ldots, v_{1(k-1)}$. The performance of the algorithm for dense problems can be improved by reapplication of vertex ordering at depths > 1. In addition, the algorithm will be faster when a good heuristic is used to determine α.

References

E. Balas and C.S. Yu, "Finding the maximum clique in an arbitrary graph", *SlAM J. Comput.* 15/4, 1054-1068 (1986).

P. Berman and G. Schnitger, *On the Complexity of Approximating the Independent Set Problem,* Springer-Verlag, Lecture Notes in Computer Science 349, 1989, 256-267.

C. Bron and J. Kerbosch, "Finding all cliques of an undirected graph", Comm. ACM 16/6, 575-577 (1973).

M. Gendreau, J.-C. Picard and L. Zubieta, *An Efficient Implicit Enumeration Algorithm for the Maximum Clique Problem,* Springer-Verlag, Lecture Notes in Economics and Mathematical Systems 304, A. Kurzhanski et al. (eds.), 1988, 79-91.

L. Gerhards and W. Lindenberg, "Clique detection for nondirected graphs: Two new algorithms", *Computing* 21, 295-322 (1979).

E. Loukakis and C. Tsouros, "Determining the number of internal stability of a graph", *lnternat. J. Comput. Math.* 11, 232-248 (1982).

P.M. Pardaios and G.P. Rodgers, "A branch and bound algorithm for the maximum clique problem", To appear in *Math. Programming* (1990).

R.E. Tarjan and A.E. Trojanowski, "Finding a maximum independent set", *SIAM J. Comput.* 6/3, 537-546 (1977).

P. Turan, "On the theory of graphs", *Colloq. Math.* 3, 19-30 (1954).

R. Carraghan and P. M. Pardalos, An Exact Algorithm For The Maximum Clique Problem, Operations Research Letters 9 (1990) 375-382 November 1990, North-Holland

Biographies

Mochamad Suyudi, is a lecturer at the Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran. Bachelor in Mathematics at the Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran, and Master in Mathematics at the Faculty of Mathematics and Natural Sciences, Universitas Gajah Mada. Currently pursuing Ph.D. program in the field of Graphs at Universiti Sultan Zainal Abidin(UNISZA) Malaysia Terengganu.

Sukono is a lecturer in the Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Padjadjaran. Currently serves as Head of Master's Program in Mathematics, the field of applied mathematics, with a field of concentration of financial mathematics and actuarial sciences.

Mustafa Mamat is a lecturerin the Faculty of Informatics and Computing, Universiti Sultan Zainal Abidin, Malaysia. Currently serves as Dean of Graduate School Universiti Sultan Zainal Abidin, Terengganu, Malaysia. The field of applied mathematics, with a field of concentration of optimization.

Abdul Talib Bon is a professor of Production and Operations Management in the Faculty of Technology Management and Business at the Universiti Tun Hussein Onn Malaysia since 1999. He has a PhD in Computer Science, which he obtained from the Universite de La Rochelle, France in the year 2008. His doctoral thesis was on topic Process Quality Improvement on Beltline Moulding Manufacturing. He studied Business Administration in the Universiti Kebangsaan Malaysia for which he was awarded the MBA in the year 1998. He's bachelor degree and diploma in Mechanical Engineering which his obtained from the Universiti Teknologi Malaysia. He received his postgraduate certificate in Mechatronics and Robotics from Carlisle, United Kingdom in 1997. He had published more 150 International Proceedings and International Journals and 8 books. He is a member of MSORSM, IIF, IEOM, IIE, INFORMS, TAM and MIM.